Attacks and Defenses Against Voice over IP (VoIP)

Abstract

Voice over IP (VoIP) has become a popular replacement for traditional telephone systems, as it provides several advantages, including cheaper long-distance charges, online conferencing, and the ability to use the work phone remotely[i]. Nevertheless, VoIP systems are subject to serious threats to confidentiality, integrity, and availability--including eavesdropping, registration hijacking, and denial of service --all of which exploit fundamental weaknesses within VoIP architecture. The mechanisms of these attacks are explored in detail, leading to a final discussion about how technology such as encryption, firewalls, and authentication, and stateless servers can be used to mitigate such threats.

Introduction / To the Community

The global Voice over IP market measured at $70.9 billion in 2013 and is projected to grow to $137 billion by 2020[ii]. Businesses are clearly and continuously adopting Voice over IP (VoIP) technology, yet a market survey shows that only 33% of customers have put or are putting a VoIP-specific security system in place[iii]. Since sensitive information is often discussed over VoIP systems, insecurity can be pricey: a company's competitive advantage may be compromised if trade secrets or upcoming products are overheard. Moreover, if client information (such as credit card information) is handled over the phone in a way that violates industry rules (such as the PCI Data Security Standard), companies may be subject to multi-million dollar fines and liability for related losses[iv]. In addition, even a small amount of voice downtime can cost a company millions of dollars in lost revenue and greater customer support costs[v]. Surely companies would rather avoid these possibilities, so it appears that there may be a disconnect between the perceived need and the actual need for VoIP security. This paper will attempt to bridge that gap by illustrating these attacks as well as possible countermeasures.

Eavesdropping: Threat to Confidentiality

Perhaps the most obvious threat, by analogue with traditional telephony, is the possibility of eavesdropping. While eavesdropping of a land-line telephone conversation requires a physical connection (also known as "wiretapping"), all that is necessary with a VoIP call is that the attacker be on the same network as the victim. That is because, like all IP traffic, VoIP traffic may be captured by a "sniffer" who receives all traffic on the network either by using promiscuous mode (on an unswitched network) or by ARP spoofing (on a switched network). In the case of VoIP, such packets are usually Realtime Transport Protocol (RTP) packets, and the reason why this attack can work is because these packets are often unencrypted by default[vi]. In other words, the captured packets contain the unscrambled voice data.

This can be easily exploited using an open-source tool such as Wireshark: the program contains a "Telephony" wizard which prompts the user to specify a signaling protocol (Session Initiation Protocol, or SIP, being the most common), which Wireshark then uses as a filter to detect all conversations that have taken place[vii]. The program will then display metadata relevant to the conversations, including who is involved (phone numbers or terminal IDs) and the current state of the call (whether it is in progress or completed). At this point, the user can simply choose a call, filter all the packets tied to that call, and use the “Play” function to reconstruct and playback the conversation.

Defense against Eavesdropping

The threat of eavesdropping can be addressed by using an encrypted VoIP protocol. Realtime Transport Protocol (RTP) packets may conform to a special profile called Secure RTP (SRTP)[viii]. SRTP is considered “a bump in the stack” in that it intercepts RTP packets and forwards an SRTP version of that same packet as the sender, while it intercepts SRTP packets and forwards an RTP version as the receiver. Its fundamental mechanism is symmetric encryption using the Advanced Encryption Standard (AES), also known as the Rijndael specification[ix], using a shared secret master key. Without this key, even a packet sniffer cannot eavesdrop because the packets will be scrambled. That places particular importance on the secure trading of the master key, which can be accomplished through a variety of mechanisms; among the most ubiquitous of these tools is a protocol created by Phil Zimmerman (of Pretty Good Privacy). ZRTP, as it is known, utilizes the asymmetric Diffie-Hellman key exchange method[x], and a closer analysis of one particular implementation of ZRTP is provided here. ZRTP is not an alternative to SRTP, but rather a tool that can serve as an integral part of a Secure RTP system.

Registration Hijacking: Threat to Confidentiality and Integrity

When considering threats posed over the phone, scammers and extortionists probably come to mind immediately. But the most dangerous call could be one that is *not* received. That is the situation brought by an attack known as registration hijacking.

To hijack a call registration is to cause incoming calls intended for a certain recipient to go to the attacker instead. This is an issue of confidentiality, since the caller trusts that the other person is who they intended to call, but it is also an issue of integrity, since the call itself is being manipulated. To understand how the attack works, one must learn how a VoIP call is started[xi]. Using packets of the Session Initiation Protocol (SIP), the caller sends an "INVITE" request containing the Uniform Resource Identifier (URI) of the intended recipient, to a proxy server. The URI is of the form "bob@voip-domain.com" or "123-456-7890@voip-domain.com." The proxy server then makes a request to another server known as the SIP Registrar, asking it for the IP address correlated with that same URI. Then the proxy server forwards the INVITE request to Bob's IP address. Now, the two participants exchange OK and ACK signals via the proxy server, at which point the call will begin[xii].

The SIP Registrar server is able to correlate the URI of the recipient with their IP address because it maintains a database of all SIP packets known as "REGISTER" requests. The headers of these packets contain several fields, including a "From" field and a "Contact" field[xiii]. The former contains the URI of the person who sent the request, and the latter contains their IP address and a parameter specifying how long the registration will last for (limits are imposed on this parameter by the server itself, which forces users to send new requests to renew their registration). These variables are where the attack takes place. An attacker can construct a REGISTER request containing Bob's URI in the "From" field, but with the attacker's own IP address in the "Contact" field. By sending this request to Bob's SIP Registrar server, the attacker can force all incoming SIP packets marked with Bob's URI to go to their own client instead.

If Bob's VoIP endpoint has not sent the REGISTER request yet, the attacker can execute a denial of service attack against him by flooding the registrar server with their own spoofed requests. This will cause the malicious binding to happen *and* prevent Bob's legitimate registration[xiv]. If Bob has already sent out the request to the registrar, the attacker must clear out that binding first. They can send a request with Bob's URI in the "From", the wildcard character (*) as the Contact, and 0 for the expiration parameter (so that it expires immediately). As far as the server is concerned, this could just be Bob deleting all registrations associated with his URI. Now the attacker can send that second request binding Bob's URI to their IP address[xv].

Defense against Registration Hijacking

There are multiple options for defense against registration hijacking. One method is to erect a firewall which permits REGISTER requests only from internal IP addresses, with the possible exception of some external ones associated with remote workers[xvi]. This scheme fails to prevent attacks from the inside— a very real possibility, given that an internal employee is more likely to know who to target for sensitive information (and what kind of sensitive information is being passed). A whitelist on the firewall could account for internal attackers, but may not be practical in workplaces where new phones or users are being added frequently. Additionally, a firewall may be spoofed by an attacker with forged source IP addresses. This motivates the use of a second method: authentication for SIP traffic.

By default, SIP packets are sent using the User Datagram Protocol (UDP) as the transport layer. UDP is used because it is lightweight and therefore fast compared to TCP, but it provides no authentication.[xvii] This is where a protocol called Transport Layer

Security (TLS) can help. When a client contacts a server that uses TLS, the server responds with a public key (as part of a public/private key system) and a certificate—a document showing that the public key belongs to that particular party. The certificates themselves are vetted and signed by a trusted third party known as a Certificate Authority, so as to prevent impersonators from sending their own public keys instead. This certificate forms the authentication mechanism of TLS. In most common uses of TLS, such as Secure HTTP, it is the server that sends the client their certificate, so that the client knows they are sending their sensitive information to the right party. That is valuable to prevent other kinds of SIP attacks (such as man-in-the-middle), but in registration hijacking, it is the *client* who needs to be authenticated, since the client is the party that could be sending a malicious REGISTER request to the SIP Registrar. Fortunately, TLS supports client certificates as an optional feature[xviii]. All that changes with SIP over TLS, at it is called, is that the server asks the client to send a certificate of its own, before processing later requests.

Denial of Service: Threat to Availability

Denial of service (DoS) attacks, in which the goal is to prevent others from using a particular application, are already well documented in the literature for situations like bringing down an ordinary website. When it comes to Voice over IP, however, there are some unique weaknesses which can be exploited, and so they merit a closer look. One of these is inherent to how VoIP signals are routed. As described before, when one party makes a call, they send an INVITE request (of the SIP protocol) to a proxy server which queries the SIP Registrar for the IP address of the intended recipient. The processing of this request is memory-intensive, and handling too many of these events can cause a proxy server to delay or even drop other requests[xix]. Thus, an attacker may flood a proxy server with INVITE requests in order to prevent it from routing legitimate requests to the SIP Registrar, thus preventing the call from starting. A slightly different flavor of this attack may target the SIP Registrar server directly by flooding it with REGISTER requests, which are also known to be expensive, preventing proxy servers from communicating with it and starting the call.

Defense against Denial of Service

Typical prevention strategies against any sort of flooding attack include powerful hardware, a firewall imposing rate limits on incoming requests, as well as server-side software that delegates each request to a different CPU thread (i.e., parallel computing), which indirectly frees memory because less needs to be stored if the processes finish more quickly. While these methods still apply in the case of SIP flooding, a more targeted approach focuses on the specific weakness being exploited—the memory-expensive processing of each request on the SIP servers. These transactions are generally expensive if they are being processed on what are called stateful servers. Stateful SIP servers store a copy of each request (as well as the request it forwards) until it receives a final reply from the next node. In some cases, they even store requests for the entire duration of the VoIP session. In contrast, an alternative as a stateless SIP server only maintains a copy of each received message as it is processing that very request. It immediately deletes the request and its context (about 3 KB each) upon forwarding a copy of it to the next node. Thus, the memory consumption is considerably reduced, along with the risk of a successful attack[xx].

Conclusion

Eavesdropping, registration hijacking, and SIP flooding both represent three of the most dangerous threats against a Voice over IP system. Encryption using Secure RTP, firewalls with carefully chosen rules, authentication using SIP over TLS, and stateless SIP servers have been proposed as countermeasures against such attacks. Fortunately, these features are available in a variety of commercial solutions. For an enterprise, the "the majority of Cisco IP phones" support both SRTP and SIP over TLS[xxi], as well as at least two models of Yealink phones[xxii]. Firewall rules may be easily enforced using any firewall (even Windows Firewall) using inbound and outbound rules against ports 5060 and 5061 (the standard SIP port for both UDP and TLS). Cisco also offers SIP servers that can be configured to support stateless mode to prevent DoS attacks[xxiii]. Even the individual consumer can equip themselves with software to support the secure VoIP protocols, using the open-source libSRTP[xxiv] for media and Kamailio packages[xxv] for signaling, respectively. Of course, there are several threats to a VoIP system that have not been described here, but these recommendations should mitigate the risk of many of those exploits as well.

[i] "Voice over IP FAQ." *Cisco*. Cisco Systems, Inc, n.d. Web. 14 Dec. 2015. <http://www.cisco.com/cisco/web/solutions/small_business/resource_center/articles/be_more_productive/voip_faq/index.html>.

[ii] Marvin, Rob. "The Rise and Fall of VoIP: Five Dos and Five Don'ts." *PCMAG*. PC Magazine, 30 Sept. 2015. Web. 14 Dec. 2015. < http://www.pcmag.com/article2/0,2817,2492221,00.asp>.

[iii] "Is VoIP Security On Your Radar? Probably Not." *Toolbox IT*. Ziff Davis, LLC, 10 June 2014. Web. 14 Dec. 2015. < http://it.toolbox.com/blogs/voip-news/is-voip-security-on-your-radar-probably-not-61674>.

[iv] VoIP Encryption in the Enterprise (n.d.): n. pag. Sonus Networks, Inc., 13 Feb. 2013. Web. 14 Dec. 2015. <http://www.sonus.net/sites/default/files/white_paper_voip_encryption_in_the_enterprise_13_february_2013.pdf>.

[v] "Mitigating Attacks in VoIP Environments." (n.d.): n. pag. Cisco. Cisco Systems, Inc. Web. 14 Dec. 2015. <http://www.cisco.com/c/en/us/products/collateral/security/ips-4200-series-sensors/prod_white_paper0900aecd801e78ba.pdf>.

[vi] Beaver, Kevin. "How to Detect and Guard against VoIP Security Vulnerabilities." *Dummies.com.* John Wiley & Sons, Inc., n.d. Web. 14 Dec. 2015. <http://www.dummies.com/how-to/content/how-to-detect-and-guard-against-voip-security-vuln.html>.

[vii] Perez, S. (2014, January 16). Getting Started with Wireshark. Hakin9 On Demand.

[viii] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <http://www.rfc-editor.org/info/rfc3711>.

[ix] Daemen, Joan; Rijmen, Vincent (March 9, 2003). "AES Proposal: Rijndael" (PDF). National Institute of Standards and Technology. p. 1. Retrieved 21 February 2013.

[x] Diffie, W., and Hellman, M. New directions in cryptography. IEEE Trans. Inform. Theory IT-22, 6 (Nov. 1976), 644-654.

[xi] Symantec. (n.d.). Retrieved December 15, 2015, from http://www.symantec.com/connect/articles/two-attacks-against-voip

[xii] Fauji, S. (2007, May 9). Session Initiation Protocol. Lecture presented in University of Maryland, Baltimore County.

[xiii] Lin, J. (2007). Security Issues and Countermeasure for VoIP. SANS Institute InfoSec Reading Room. Retrieved December 15, 2015, from https://www.sans.org/reading-room/whitepapers/voip/security-issues-countermeasure-voip-1701

[xiv] Fauji, S. (2007, May 9). Session Initiation Protocol. Lecture presented in University of Maryland, Baltimore County.

[xv] Collier, M. (2005, June 1). VoIP Vulnerabilities -- Registration Hijacking. Retrieved December 15, 2015, from http://download.securelogix.com/library/Registration_hijacking_060105.pdf

[xvi] Collier, M. (2005, June 1). VoIP Vulnerabilities -- Registration Hijacking. Retrieved December 15, 2015, from http://download.securelogix.com/library/Registration_hijacking_060105.pdf

[xvii] C. Shen, E. Nahum, H. Schulzrinne and C. P. Wright "The Impact of TLS on SIP Server Performance", Proc. 4th Annual ACM Conference on Principles, Systems and Applications of IP Telecommunications (IPTCOMM), pp.63 -74

[xviii] Hess, A., Jacobson, J., Mills, H., Wamsley, R., Seamons, K., and Smith, B. 2002. Advanced Client/Server Authetication in TLS. In Network and Distributed System Security Symposium. San Diego, CA.

[xix] G. Ormazabal, S. Nagpal, E. Yardeni and H. Schulzrinne "Secure SIP: A Scalable Prevention Mechanism for DoS Attacks on SIP Based VoIP Systems", Proc. 2nd International Conference on Principles, Systems and Applications of IP Telecommunications (IPTComm), pp.107 -132

[xx] S. Ehlert, "Denial-of-service detection and mitigation for SIP communication networks," Ph.D. dissertation, Berlin Institute of Technology, 2009. [Online]. Available: http://opus.kobv.de/tuberlin/volltexte/2010/2496/

[xxi] Cisco IP Phone Certificates and Secure Communications. (n.d.). Retrieved December 15, 2015, from http://www.cisco.com/web/about/security/intelligence/IP_Phone_Security_WP.html

[xxii] Supported Business Phone Equipment | Fonality. (n.d.). Retrieved December 15, 2015, from http://www.fonality.com/features/equipment-software-only

[xxiii] Cisco SIP Proxy Server Version 1.0 Administrator Guide - Configuring the Cisco SIP Proxy Server [Cisco SIP Proxy Server]. (2007, May 12). Retrieved December 15, 2015, from http://www.cisco.com/c/en/us/td/docs/voice_ip_comm/sip/proxies/1-0/administration/guide/ver1_0/config.html

[xxiv] Libsrtp: A library for secure rtp. (n.d.). Retrieved December 15, 2015, from http://srtp.sourceforge.net/srtp.html

[xxv] Features | Kamailio (OpenSER) SIP Server. (n.d.). Retrieved December 15, 2015, from http://www.kamailio.org/w/features/