An Analysis of Security Threats and Tools in SIP-Based VoIP Systems

Jeffrey Albers, Bradley Hahn, Shawn McGann, Seungwoo Park, Rundong Zhu

Jeffrey.Albers@colorado.edu Bradley.Hahn@colorado.edu
Shawn.McGann@colorado.edu Seung.Park@colorado.edu
Rundong.Zhu@colorado.edu

A capstone paper submitted as partial fulfillment of the requirements of the degree of Masters of Interdisciplinary Telecommunications at the University of Colorado, Boulder, April 29, 2005. Project directed by Professor Douglas Sicker.

Abstract

Voice over Internet Protocol (VoIP) is subject to many security threats unique to both telephony and traditional Internet data transmission. As adoption of Session Initiation Protocol (SIP) based telephony increases, concerns are rising over risks to system confidentiality, integrity and availability. Currently, several VoIP security tools are available to detect vulnerabilities and protect against attacks. In this section one of this paper, known vulnerabilities to VoIP networks are compiled. In section two, several software and hardware products are evaluated to determine effectiveness at mitigating security risk. All evaluated tools were found to have serious flaws that limit usefulness. Users should therefore employ these tools with caution, realizing most tools are still in developmental infancy.

1 Introduction

Security tools such as protocol analyzers, vulnerability assessment utilities and security monitoring utilities are the primary tools in a security professional's arsenal. These tools have reached a high level of dependence among security professionals for protecting operating systems, network protocols and other applications from attackers. While the security tools for developed technologies are strong, it remains difficult to develop comprehensive security tools in the early stages of the life cycle of an emerging technology. Voice over Internet Protocol is an example of such a technology [1]. This paper explores the known vulnerabilities and tests several open source and commercial Session Initiation Protocol (SIP) based VoIP security tools to determine what issues need to be addressed in the future development of VoIP security tools.

On the Internet, popular applications tend to become popular targets of attackers. Networking protocols, operating systems, web browsers, email clients and other applications are examples of pervasive targets that have suffered from this curse. VoIP presents a likely next popular target because of its growing popularity. Furthermore, VoIP presents new challenges in that it differs from traditional Voice (i.e., PSTN) in a number of ways. For example, no single entity controls the development and monitoring of the network and voice applications in a VoIP system. Implementers and end users are empowered to configure their own systems however they see fit. This leads to the issue of possible user misconfiguration, which is a serious security threat in any application. Other security concerns within VoIP include potentially poor software development, which could lead to various security problems.

While there have been little wide spread attacks unique to VoIP systems as of yet, the potential exists for security issues to have a detrimental effect on the success of the technology. People have become accustomed to the 99.999% availability rate standard on the PSTN, and many will likely expect VoIP to meet that service level [2]. A security breach that compromises VoIP availability could be detrimental to the public confidence in the technology - further establishing the need for high-quality VoIP security tools [3].

Many companies and open source groups have already begun tailoring security programs such as vulnerability assessment tools, intrusion detection/prevention systems and firewalls to meet the requirements for VoIP. This paper shows that the functionality of current tools is severely limited and should not be overly relied on to properly secure a VoIP implementation. As the research concludes, there are a significant number of vulnerabilities that the tested fail to address. These issues must be resolved before VoIP security tools can be considered valid security utilities and included in a security professional's arsenal.

2 Vulnerabilities

The vulnerabilities in VoIP encompass not only the flaws inherent within the VoIP application itself, but in the underlying operating systems, applications, and protocols that VoIP depends on. The complexity of VoIP creates a high number of vulnerabilities that affect the three major areas of information security: confidentiality, integrity, and availability. For purposes of organization, the vulnerabilities that affect these three major areas have been separated based on the layers of the TCP/IP networking model. Those layers are: network interface layer, network layer, transport layer, and application layer.

The vulnerabilities in the sections below concentrate on those that can be mitigated or monitored by tools. Several aspects of network security have been omitted from this list of vulnerabilities because they are outside the relevance of paper. Non repudiation, access, and accounting have been left out of the vulnerabilities section despite their fundamental importance of network security. Our research shows that many of the vulnerabilities affect more than one area of information security and often include confidentiality, integrity and availability weaknesses. Table 1 shows the relationship among the individual vulnerability and which areas of network security they affect.

Physical security is a major issue in all information systems, VoIP included. However, it is very difficult for a tool to assess or monitor the status of physical security. VoIP implementers should still consider physical confidentiality risks. While most attacks exploit weaknesses within one or more of the networking layers, some are also dependent on physical attack vectors that exist in unutilized interfaces on the VoIP equipment. This includes data jacks, switch/hub ports, wireless range, and additional interfaces on the VoIP phone (i.e., a built-in hub). These interfaces should remain disabled unless they become necessary for functionality [4]. Furthermore, security measures such as authentication, address filtering, and alarms for when devices are disconnected can mitigate the risks involved in physical security.

Table 1: VoIP confidentiality, integrity and availability vulnerabilities examined at the various protocol layers.

	Vulnerability	Confidentiality	Integrity	Availability
Data Link	Physical Attacks	x		x
	ARP cache	x	x	x
	ARP flood			x
	MAC spoofing	x	x	x
Internet	IP spoofing			
	Registration server, IP phone, MGCP, DNS, etc	x	x	x
	Redirect via IP spoof	x	x	x
	Malformed packets	x	x	x
	IP frag	x	x	x
	Jolt			x
Transport	TCP / UDP flood			x
	TCP / UDP replay	x	x	
Application	TFTP server insertion		x	
	DHCP server insertion (redirect)		x	
	DHCP IP address starvation			x
	ICMP flood			x
	SIP			
	Registration Hijacking	x	x	x
	Call Hijacking (MGCP NotifiedEntitiy parameter)	x	x	x
	Message body modification	x	x	
	RTP insertion			
	Spoof via header	x	x	x
	Cancel / bye attack			x
	Malformed method			x
	Redirect method	x		x
	RTP			
	SDP redirect			x
	RTP payload			x
	RTP message tampering	x	x	x
	Encryption	x	x	x
	Default settings / passwords	x	x	x
	Disable unnecessary services HTTP, FTP, etc	x	x	x
	Buffer overflow	x	x	x
	Legacy Network Interaction	x	x	x
	DNS Availability			x

2.1 Confidentiality

Confidentiality is a major issue in VoIP when compared to the traditional PSTN. Confidentiality refers to the protection of data from being read by an unauthorized user. Historically, an attacker would need physical access in order to eavesdrop on a conversation or illegitimately access the telephone service. Since VoIP uses open protocols and public networks, a number of attack

vectors exist which the proprietary nature of the PSTN had previously protected access to the media and signaling.

2.1.1 Network Interface Layer

There are several logical exploits at the network interface layer that could potentially result in a system confidentiality breach. Most of these attacks require a compromise at another layer such as physical access to the system or broken authentication. Once the threat agent has gained access to the network, it is possible for them to perform a number of attacks. Media Access Control (MAC) address spoofing at the network interface layer includes impersonating the registry server, gateway, proxy, user agents, or other devices. This kind of attack can compromise the privacy of the conversations that transverse the network, or allow an illegitimate user to place VoIP phone calls. However, due to ARP, this kind of spoofing is more likely to happen at the network layer and will be analyzed later in this paper. MAC spoofing will more likely be an attack vector on networks that have MAC address filtering, or used to help an attacker to cover his tracks [4].

Another vulnerability to confidentiality in the network interface layer deals with an Address Resolution Protocol (ARP) flood attack. An intruder who has gained access to the VoIP network can send ARP commands, which can corrupt the ARP caches of legitimate devices and their traffic. Blindly flooding a network with ARP replies can corrupt the ARP cache of a device. If done correctly, an attacker can use this method to re-route traffic to intercept data and voice packets. [5]. Limiting physical access and strong authentication mechanisms can reduce the risk of attacks on confidentiality at the network interface layer.

2.1.2 Network Layer

Network layer confidentiality vulnerabilities stem from the Internet Protocol (IP) itself. The most prevalent vulnerability in IP is address spoofing. In a VoIP environment, an attacker can impersonate many different devices by spoofing those devices' IP addresses. These devices include: registration servers, SIP proxy servers, IP phones, Media Gateway Control Protocol (MGCP) servers, and Domain Name Server (DNS) servers. By impersonating a valid user of IP phone through spoofing, an attack can use the VoIP system to make unauthorized calls.

In a VoIP environment, a threat agent can decipher the IP address of a phone simply by calling that number or extension and sniffing the packets that it receives. While simply knowing the IP address of a foreign phone does not constitute a true vulnerability, it can lead to other attacks. Knowing the IP address of a phone can allow an attacker to specifically target that phone for eavesdropping or spoofing.

One method for eavesdropping on conversations involves spoofing the default gateway's IP address on a phone to redirect all VoIP packets to an attacker's machine. With traditional IP forwarding, this kind of attack can remain hidden from intrusion detection systems. IP phones with remote administration capabilities severely increase the risk of this type of attack. Implementing network layer filtering through the use of a firewall can reduce the risk of this kind of attack [5].

2.1.3 Transport Layer

Vulnerabilities at the transport layer primarily involve Real Time Protocol (RTP). An attacker can intercept RTP packets and use a packet sniffing application can capture entire RTP sessions. An attack tool known as Voice Over Misconfigured Internet Telephones (VOMIT) takes

advantage of this vulnerability. VOMIT captures VoIP packets in transit, and then converts the captured data into .wav files that can be listened to on any modern digital multimedia player. This kind of attack presumes access to the physical network carried the VoIP traffic. While completely preventing this kind of attack may prove to be impossible, the use of encryption like IP Security (IPSec), Secure Real Time Protocol (SRTP), or Transport Layer Security (TLS) can greatly reduce the risks of such an attack [6].

The introduction of encryption to the VoIP system is not a solve-all solution when it comes to VoIP security. While not countering flaws such as buffer overflows, malformed packets, replay attacks, or physical security; encryption also can introduce new vulnerabilities into they system. For instance, TLS forces the endpoints to exchange keys. An attacker can intercept the key exchange, capture the encrypted packets during the communication, and then decrypt the packets. Or the attacker could run a typical man-in-the-middle attack if they can break the communication during the key exchange of TLS [7].

2.1.4 Application Layer

Application Layer vulnerabilities Media Access Control (MAC) address spoofing involve the SIP protocol itself. In one type of attack, spoofing is employed to impersonate a SIP proxy server. User agent requests can then be intercepted and modified to do any number of malicious things. This kind of attack can be mitigated through the use of server authentication [8].

The SIP protocol does specify certain standards for authentication. The original SIP protocol's authentication method transmitted the username and password in clear text over the network. Under the current specification, SIP includes authentication via Hypertext Transport Protocol (HTTP) digest. Although this is a relatively strong form of authentication, it does not measure up to standards such as Public Key Infrastructure (PKI) due to poor key management and lack of a third party authority. Additionally, SIP includes a Secure / Multipurpose Internet Mail Extensions (S/MIME) mechanism for encryption. This mechanism is vulnerable to a man-in-the-middle attack where a threat agent can intercept the key during the beginning of an exchange [9].

Registration hijacking is another common and altogether simple attack on sip-based VoIP systems. "The SIP registration mechanism allows a user agent to identify itself to a registrar as a device at which a user is located. A registrar assesses the identity in the FROM header field of a REGISTER message to determine whether this request can modify the contact addresses associated with the address-of-record in the TO header field. The FROM field of a SIP request, however, can be modified arbitrarily by the owner of User Agent (UA), and this opens the door to malicious registrations [8]." This can result in attackers gaining the ability to place calls over the VoIP system or redirecting legitimate class to a malicious user's device.

Call hijacking is an application layer attack that takes advantage of flaws in SIP. SIP's design includes REDIRECT messages that allow a user to move from environment to environment (i.e., from landline to wireless phone) while maintaining a conversation on a single call. In a call hijacking attack, the attacker sends a REINVITE message that can reroute the RTP flow to the attacker's IP address. Part of this attack serves as a denial of service attack for the caller spoofed by the attacker, as they would lose their connection to the call. The recipient of the REINVITE message would then be directly connected to the attacker, who can hear anything that the recipient says. The result is a breach of confidentiality and privacy [10].

SIP also supports Instant Messaging (IM), which is subject to a different type of security vulnerability. An attacker can falsify the header of an IM packet in order to send an instant

message that appears to have been sent by a legitimate user. These messages can contain links to malware or request confidential information (known as phishing) [10].

MGCP has a built in NotifiedEntity parameter that allows endpoints to change MGCP servers. An attacker can exploit the NotifiedEntity parameter in a similar manner call hijacking attack that exploits the SIP REINVITE message vulnerability described above. Here the attacker can use the NotifiedEntity parameter to redirect the endpoints to a false MGCP server, thus hijacking the communication between the endpoints [6].

Nearly all modern networking equipment is shipped with a default login, password and configuration. These initial settings are relatively well documented and available to the public. This information can be used for a number of exploits, which will be explored further in subsequent sections. With respect to confidentiality, knowing the default login/password on a switch can allow an attacker to reconfigure the device to mirror packets from port to port. This creates a situation where packets can be intercepted without setting off any alarms in the system. Default settings should be changed and/or disabled wherever possible, and disabling port mirroring should be examined as well. Furthermore, remote administration capabilities should be either disabled or enabled with strong encryption to prevent the capturing of configuration details or login/password by malicious packet sniffers. With so many devices implementing HTTP servers for both local and remote configuration (including switches and VoIP terminals) eliminating default logins becomes an increasingly important area to recognize. HTTP traffic can be sniffed and read by anyone with access to the local network segment. HTTP servers should be appropriately disabled. If it is necessary for the device to have an HTTP server installed it should have strong encryption enabled such as Secure Sockets Layer (SSL) [5]. Enabling encryption also applies to other remote applications used for remote administration such as telnet and programs used for provisioning like File Transfer Protocol (FTP) and Trivial File Transfer Protocol (TFTP) [6].

Provisioning in itself contains vulnerabilities that could potentially compromise the system. Provisioning info is typically stored in the unencrypted eXtendible Markup Language (XML) format. The XML data is extremely sensitive and can be intercepted while in transit or if the provisioning server is breached. “Provisioning information could include passwords, server and other reference addresses, parameters for enabling/disabling test environments etc [6].”

Soft clients introduce many vulnerabilities into the system that would not otherwise exist with dedicated IP phones [6]. The vulnerabilities introduced by soft phones stem from two major sources. First, by converging voice and data on the same network, the vulnerabilities in each individual system propagate to the other. The voice network inherits the vulnerabilities of all the servers, operating systems, switches, routers, and other applications on the data network, while the data network inherits the vulnerabilities in the VoIP architecture as described in this paper. A compromise of one part of the network induces a compromise of the other. This unification increases the risk, probability of attack and losses incurred in the event of a successful attack [11]. Secondly, soft phones can fall victim to the operating system that it runs on and bundled applications therein. Operating systems, as with other software, contain many programmatic flaws such as poor memory management that can lead to buffer overflows. A compromise of the operating system or application can result in a denial of service scenario. Worse, full control of the system could be gained, allowing access to confidential data [7].

Buffer overflow attacks can occur against any application without proper memory management controls. Many VoIP implementations, both open source and commercial, likely contain such flaws. These holes can be discovered by the application developers or by hackers.

To address buffer overflow holes in programming, it is imperative to keep all applications and operating systems on the network up to date with the latest service packs and patches. Every software component within the system increases the likelihood that a buffer overflow vulnerability exists in the system. Disabling the services unnecessary in the designed function of the system can help alleviate the buffer overflow threat. Furthermore, ensuring that necessary services are not run with high system privileges can help to minimize damage from a breach.

In addition to inherited threats, VoIP systems are also subject to other attacks that may seem peripheral to the VoIP system, but pose a direct threat nonetheless. Weak passwords are subject to dictionary and brute force attacks that can yield unauthorized access into the VoIP system [8]. From there the attacker can access system resources, confidential data, plant malware and change configurations Media Access Control (MAC) address spoofing. Malware in general is a concern for VoIP systems. There are many viruses and worms that propagate through public networks and VoIP systems may be built on technology vulnerable to these agents.

2.2 Integrity

Integrity is difficult to protect in VoIP systems. A breach in integrity includes the unauthorized modification or deletion of voice/data content, passwords, configuration, and other information stored within the system. As in many network systems, VoIP applications use encryption, hash algorithms and message digests to aide in maintaining integrity. Although encryption can alleviate the threat of attacks on VoIP system integrity, there are still fundamental vulnerabilities in VoIP systems that can exploit integrity. The success of an attack on integrity often leads to breaches in confidentiality and availability as shown in the other sections of this paper. Many of the other vulnerabilities listed in the other sections of this paper. The nature of data places most of the vulnerabilities in the upper layers of networking model. The majority of the vulnerabilities are found in the transport and application layers, with few in the lower layers.

2.2.1 Network Interface Layer

As mentioned in confidentiality section, VoIP systems are susceptible to ARP cache flooding attacks. While the result of the attack is the redirecting of a VoIP conversation, the attack also affects the integrity of the sender's ARP Table.

2.2.2 Network Layer

Integrity can be compromised at the network layer by an IP address spoof. In the absence of higher-layer authentication protocols, attackers who spoof an IP address of a legitimate device may be able to alter data on that device and possibly throughout the system [4].

2.2.3 Transport Layer

The basis for an insertion attack deals with vulnerabilities at the transport layer. In a standard man-in-the-middle attack, an attacker can capture packets as communicating agents send messages to and from one another. The attacker can then forward the TCP or UDP packets as is, or alter the packets by deleting information and/or including new information. Furthermore, VoIP is vulnerable to a TCP or UDP replay attack, where an attacker captures a full session and relays that message one of the communicating agents (or a third party) after the original session has ended.

2.2.4 Application Layer

The majority of the threats to integrity in a VoIP system reside in the application layer. A number of vulnerabilities exist that allow an attacker to change the configuration of a device within the VoIP system. The first of these vulnerabilities is a DHCP server insertion attack. An attacker can implement a rogue DHCP server in a network and when a phone or UA makes DHCP request, the rogue server replies with false information. This attack alters the IP address, DNS servers and routing table of the DHCP client [5]. An attacker can perpetrate a similar attack through a rogue TFTP server. In a TFTP server insertion attack, when an IP phone requests configuration information from a TFTP server, the rogue server replies with a false image. A successful TFTP insertion leads to a complete corruption of the IP phone's configuration [6]. In both insertion attacks IDS, authentication, and address filtering can mitigate the risk.

Several integrity vulnerabilities exist in the SIP protocol as well. One example is a SIP modification attack. "Modification attacks occur when an attacker intercepts the signaling path and tries to modify SIP messages in order to change some service characteristics. For example, this kind of attack can be used to hijack the signaling flow forcing a particular route, or to change a user registration or modify a service profile [12]." SIP is also vulnerable to an attacker modifying the message body. Attackers can change session encryption keys, MIME bodies, SDP and encapsulated telephony signals within SIP. This attack can allow a threat agent to redirect an RTP stream or eavesdrop on a conversation. Information in the header fields is also subject to modification attacks [8]. The previous section of this paper mentioned SIP registration hijacking as a vulnerability to confidentiality. However, the registration server's primary threat is to integrity. As stated earlier, this attack and lead to false registrations within registry, a classic example of compromised data integrity [6].

VoIP's reliance on RTP subjects it to an insertion attack, whereby an attacker inserts messages into the RTP stream delivered to one of the users. This kind of attack allows an attacker to insert indecent words or other noises into a legitimate conversation [5]. Captured RTP packets inserted into another RTP stream serves as a kind of replay/insertion attack to impersonate a legitimate user.

User authentication presents another attack vector in VoIP systems. Access to legitimate user accounts provides a way for an attacker to read, delete, or modify data or configurations. Weak, default and blank passwords make it easy for an attacker to gain access to accounts.
As in all areas of network security, poorly written software is a major risk to VoIP integrity. An exploited buffer overflow vulnerability can, allow an attack to modify data and configurations.

2.3 Availability

Availability assures that storage and transportation facilities for an information system are accessible to authorized users at the time they require access. Attacks directed at availability are generally categorized as a denial of service and can be classified into four broad types: bandwidth consumption, resource starvation, routing attacks, or programming flaws [6].

Bandwidth consumption attacks generally comprise of flooding the network with a specific type of traffic. The quantity of the flooding traffic eventually consumes enough bandwidth to significantly reduce or eliminate the path for legitimate traffic. A single machine typically cannot produce the amount of traffic required to degrade a link, therefore these attacks are generally conducted with groups of computers under the control of a hacker through the use of a virus/worm. Preventive measures typically include operating an Intrusion Detection / Prevention System (IDS/IPS) to identify and mitigate the attack [6].

Resource Starvation attacks flood a device (opposed to links in a bandwidth consumption attack). The attacker generally initiates a large quantity of requests to a device so that resources are exhausted and service is not available to other users. Preventive measures are similar to the actions taken for bandwidth consumption attacks [6].

Routing Attacks involve manipulating routing information or protocols in order to intercept / interrupt legitimate traffic. The method typically involves spoofing data in legitimate packets or inserting entirely false packets. Preventive measures generally consist of protecting devices though physical security and encryption.

Programming Flaws are unintended bugs in software that can be exploited by a malicious user in order to gain access to a system. These flaws are typically attributed to sub-standard programming as a result of rushing software to the market. Preventive measures are typically reactive in that they rely on installing software patches [6].

2.3.1 Network Interface Layer

At the Network Interface Layer, attacks affecting availability can be separated into two broad categories: physical security and OSI Layer 2 functions. Physical security protects devices on the network by restricting access. This protection prevents an attacker from inflicting physical damage to a device through actions such as cutting links or removing power. Security tools can do little to identify or prevent these types of attacks.

In reference to OSI Layer 2 attacks, the most common exploit is via ARP. The two primary attacks are the ARP flood [13] and ARP cache [13][14] attacks. ARP flooding consists of overwhelming a device with spoofed ARP replies in order to overflow the ARP cache. The effects of this type of attack vary from bandwidth consumption on links to crashing the system by overwhelming the processor. In the ARP cache attack the aggressor exploits the stateless property of ARP. Since ARP is a stateless protocol, devices will update their ARP cache whenever they receive an ARP reply - regardless if the information was requested. As a result, an attacker can manipulate the ARP cache on a device by sending forged ARP replies. Attacker modifications typically involve the insertion of a non-existent address in order to black hole traffic or the insertion of an attacker specified addresses in order to redirect communications [5].

2.3.2 Network Layer

Availability as well as confidentiality are affected at the network layer predominately through spoofing IP addresses of network devices in order to intercept/redirect traffic. The used of malformed packet attacks are also common at the network layer. The IP phone netmask vulnerability is in many ways similar to the ARP cache issue described earlier [5]. In this exploit, an attacker can change the IP address and mask of a router in order to redirect packets to a device of his choice.

The two most common malformed packet attacks are the IP fragmentation attack and "jolt" attack. In the IP fragmentation exploit, the attacker creates large packets that will be broken apart before they are sent to the recipient. During reassembly at the receiver an overlap in the data typically complicates the procedure. The results can vary from resource starvation to a system crash [15]. Similarly, in the jolt exploit an attacker sends malformed IP packets to the IP processing stack. Again, the overlap in data results in a processing error or unstable state.

2.3.3 Transport Layer

Vulnerabilities affecting availability at the transport layer typically involve flooding or fragmentation of TCP and UDP. In the TCP SYN flood, an attacker directs malicious TCP connection requests at a server. The malicious requests fail to complete the three-way handshake and result in a half open connection. As subsequent forged requests create additional half open connections, the server eventually exhausts its memory and crashes [16]. Similarly in a UDP flood, UDP packets are directed at a random port on a device. Once the receiver realizes the socket does not exist it generates an ICMP destination unreachable packet. As the attacker continues to send these malicious packets, an endless loop is created which eventually consumes enough memory to crash the system [16].

2.3.4 Application Layer

Availability is most affected at the Application Layer. The majority of vulnerabilities involve SIP, however there are numerous other applications that are exploitable. In SIP, an attacker can execute a DoS attack by spoofing the IP address and Via header of a request. The attacker inserts information belonging to the victim into a request then sends the request to numerous SIP UA's or proxies which generate traffic to the victim. In the SIP CANCEL/BYE exploit, an attacker can end a call by directing a spoofed "CANCEL" or "BYE" message to a device [10]. Similarly in an ICMP port unreachable attack, the attacker can end a call by directing a spoofed "ICMP Port Unreachable" message to a device [17]. Furthermore, an attacker can impersonate a user agent during the SIP registration process or during a REDIRECT method in order to change user agent information [10]. A successful attacker can potentially remove the contacts for a given URL and replace them with his own contact information in order to intercept or redirect calls. This attack is often referred to as call hijacking [8].

Availability can also be greatly affected by exploiting SDP and RTP. For example, an attacker can modify SDP data so that RTP media streams are redirected to a wiretapping device. The RTP data can then be used for a man-in-the-middle, redirect or replay attack. In addition, in the RTP payload exploit, an attacker can send RTP packets filled with random bytes (both the header and the payload) to corrupt the jitter buffer in a device. This exploit generally results in intermittent conversation or a system crash [10].

RTP exploits can also involve insertion and message tampering. A common attack involves exploiting RTP packet numbering [10]. In RTP, the function of packet numbering is not to assemble packets in order - it is only to alert higher layers of a problem. Since RTP does not mandate that packets arrive in order, an attacker can insert malicious packets by using a reasonable packet number captured from a recent packet. From here various Denial of Service attacks could be initiated.

In an ICMP (Ping) Flood an attacker directs numerous ICMP echo request messages at a device (generally through a broadcast to other devices) [16]. The replies to these requests are directed at the victim and eventually overwhelm the system processor or consume enough bandwidth to create a denial of service. In the DHCP IP Address Starvation exploit an attacker forges enough DHCP requests to lease all available IP addresses from a DHCP server. As a result, there are no addresses for legitimate requests [18].

Other attacks include SPIT (Spam over Internet Telephony). Similar to SPAM, individuals automate the delivery of millions of advertisements to the voice mail of potential customers. Significantly larger in size than SPAM email messages, the stored sound files associated with SPIT create a much larger burden on network resources. In an account lockout attack, a

malicious user can create a denial of service by incorrectly authenticating to a device in order to trigger the automatic lock out system. This act prevents legitimate users from using the device. In order to prevent dictionary attacks counters are used to tally the number of incorrect password attempts. In this exploit, an attacker simply executes the required number of incorrect attempts in order to lock out a user [5].

3 VoIP Security Tools

Some of the lessons that can be taken from the growth of the Internet show that all security concerns cannot be realized upfront and exploits grow dramatically as the number of target systems multiply. As VoIP systems become more prevalent and risk grows, network engineers need to make sure the proper precautions are taken to prevent security breaches. While no significant exploitations of VoIP systems have been documented to date, researchers and vendors are developing new security tools designed specifically for VoIP in an attempt to stay ahead in a game that has only just begun. However, up until this point no attempt has been made to evaluate the capabilities of current VoIP security tools. The following sections contain an appraisal of testing tools and active network appliances and an assessment of their ability to detect and prevent potential security threats.

Several commercial and open source testing tools claim to be useful in securing VoIP systems. SiVuS and the c07-sip tests for PROTOS are freely available programs for SIP robustness testing. Software is robust if it is able to withstand exceptional input and stressful conditions. These tools essentially work by injecting exceptional elements into SIP protocol messages. An exceptional element consists of some abnormality that would not normally be found in a SIP packet, such as a large number of characters or an IP address in an unrecognizable format [19]. If a VoIP software engineer made a mistake in coding the program, checks that verify signaling data fields are of the correct size and format may be missing or in error. When the user agent or server does not handle the exceptional elements correctly, they can cause the program to hang or crash. The results of an exceptional element attack can range from service denial to unauthorized access. Thus, it is important for software engineers to run thorough conformance and robustness checks against VoIP software.

Tests that verify the robustness of VoIP products can also be useful to network engineers. It should not be assumed that SIP implementations are free of malformed packet vulnerabilities. In a 2003 robustness survey using the c07-sip test cases, Weiser and Laakso found nearly all implementations tested to be vulnerable to several exploits that result in denial of service [20].
The analysis of robustness testing tools below includes three freely available testing programs: SiVus, PROTOS c07-SIP Test Suite, and SIP Forum Test Framework (SFTF).

3.1 Testing Methods

The documentation of each security program was examined to determine recommended use. If no such information could be found, assumptions were made about how a standard scan might be conducted. An Asterisk PBX running on Debian Linux was used as the test subject when the program was capable of testing registrar/proxy servers. SJPhone (on Windows XP) and Linphone (on Fedora Core 3) were the two soft phones used for user agent testing. The programs were evaluated in terms of robustness, ease of use, documentation, usefulness and ability to live up to developer claims of functionality. Since all products have different

functionality, a direct comparison on all grounds was not feasible. Instead, the evaluation was based on the strengths and weaknesses of each tool at a more general level.

3.2 SiVuS

SiVuS claims to be the first publicly available vulnerability scanner for VoIP networks. The group at vopsecurity.org released the first version in October of 2004 and continues to develop it. The program, developed for Microsoft Windows, contains three components. The first is the SIP Message generator, which can be used to test issues or generate demonstration attacks. Second, the SIP component discovery is useful for identifying targets for analysis. Finally, the SIP vulnerability scanner can be used to verify the robustness and security of SIP phones, proxy servers and registrar servers [21].

3.2.1 Strengths

The SiVuS scanner's main strength is its Windows-based GUI design, which makes it more user friendly than several of the other vulnerability scanners tested. Reports are generated in an easy to read html page. While this report does not contain all of the information necessary for proper evaluation by a security engineer, the results are in a format easier to view than many of the command line based scanners. The SiVuS scanner also checks both the robustness of all SIP message types and for the presence of several security features.

3.2.2 Weaknesses

One of the greatest weaknesses of SiVuS is a lack of information in its reports to analyze the test output properly. For example, when testing the Asterisk PBX, 281 of 360 checks were reported as "high" risk. The reports did not indicate which tests passed, nor did they offer any indication as to what qualifies as a "pass." Without knowing what makes the program report a failure, it is impossible to know what to fix or even if the program is reporting actual vulnerabilities. For instance, the program could be looking for a certain response packet to each malformed packet within a timeout period. If the server does not respond because it is configured to ignore malformed packets, this could be acceptable behavior and not indicative of any vulnerability. Furthermore, the user guide lacked key information as to how the tests work or what to do if a test fails. No documentation was given as to what the options in the program accomplish or how a typical user would use the program. Since SiVuS is the first program of its kind, it is all the more critical that the documentation be updated so users can understand the program's purpose. The report, scanning activity log and packet sniffer logs were all examined in an attempt to get a complete idea of the scanner's operation. If all three of these views were integrated, understanding the program output would be much easier.

Testing revealed that the SiVuS program contains many bugs that may result in frustration or deception to the user. First off, the SIP device scan feature failed to locate the Asterisk server or the two soft phones on the test network. A user could potentially miss a device with a security threat when using this function. Secondly, several issues arose that required SiVuS to be restarted. For example, saved configurations and the file name in "log all scanning activity" are only loaded when the program is initialized. In addition, when the user cancels a test in progress, the program must occasionally be restarted before another test can be run. SiVUS cannot recover from errors such as "Could not bind to port 5060" without restarting. Third, two test cases involving authentication were found to report inaccurate results (see Appendix A). Fourth, running the test cases repeatedly fails to find a target on the first attempt, but succeeds when the

user runs the test again. Lastly, while the TLS option in the configuration page is deactivated and the user guide says it is not ready for this version, the first error displayed on the activity log suggests that the program does indeed try to connect via TLS and gets a connection refused error.

3.2.3 Developer Claims and Analysis

In the user guide [21], the developers claim the program is capable of:

1. Analysis of the SIP message headers to identify vulnerabilities such as Buffer overflows or denial of service attacks. These checks can be selected and configured with variable values, by the user.
Analysis: SIP message headers are not analyzed, but rather the protocol implementations are analyzed for robustness. User defined tests cannot be preformed with authentication, limiting their usefulness.

2. Authentication of signaling messages by the SIP component under analysis.
Analysis: The test for checking INVITE authentication requirements incorrectly reported a test failure (see Appendix A).

3. Authentication of registration requests.
Analysis: The claim was verified through testing; however, an error was found in the report (see Appendix A).

4. Inspection for secure communications (SIPS) and encryption capabilities.
Analysis: The test equipment only worked with UDP. Since SIPS can only be used with TCP, this claim could not be verified.

In conclusion, the number of significant bugs and lack of documentation currently limit the usefulness of the SiVuS program to the network engineer. However, the program represents a significant contribution in the development of a VoIP vulnerability scanner that has the potential to one day prove quite valuable.

3.3 PROTOS c07-SIP Test Suite

The PROTOS program was developed at the University of Oulu in Finland as an inexpensive way to test protocol implementations for security robustness. The c07-SIP test suite was designed for an initial survey of SIP User Agent and server implementations in 2003.
The PROTOS tool contains over 4,500 test cases that inject exceptional elements into SIP INVITE messages, including SDP. Monitoring the target SIP implementation device for abnormal functionality is necessary to determine test results. The c07-SIP initial trial defines a test failure as occurring when: (1) “A device undergoes a fatal failure and stops functioning normally; (2) a process or a device crashes or hangs and needs to be restarted manually; (3) a process or device crashes and restarts automatically; or (4) a process consumes almost all CPU and/or memory resources for an exceptionally long or indefinite time.” [22]

3.3.1 Strengths

One of the great strengths of this program is its simple design. There are no reports or logs to interpret as the user can observe the client to see if a service denial occurred. The documentation

is detailed enough for the network engineer to get an effective understanding of the capabilities and functionality of the program. All of the test cases are outlined in a table leaving little question as to what every case does. The source code is available as a reference if more in-depth knowledge is needed or for further application development. Finally, the test cases were designed using a comprehensible methodology.

3.3.2 Weaknesses

The most significant weakness of the c07-suite is the scope of its test cases as it only covers INVITE messages. The program also does not appear to be compatible with the Linux operating system (see Appendix B), even though it was supposedly developed in Java to be cross platform. Finally, the lack of a report can make it difficult to determine test results. For example, locating the test where a server crashed is a time consuming process of limiting the test cases and repeating until only the single case that caused the crash is run.

3.3.3 Developer Claims and Analysis

In the paper "Security Testing of SIP implementations," the developers claim the program is designed to:

1. "[E]valuate implementation level security and robustness of Session Initiation Protocol (SIP) implementations [20]."

Analysis: Our tests showed that the program was indeed effective at identifying certain serious robustness issues. However, the limited scope of the tests (INVITE messages only) means that the tests are far from comprehensive in testing security and robustness. A better claim for the developers to make would be that it helps identify robustness issues in INVITE message processing.

The developers realized the main weakness of the test suite stating that, "A more comprehensive test-suite should be developed as the SIP scene matures [20]." Codeomicon Inc. has further developed the PROTOS tool into a commercial test tool that includes a graphical user interface, PSTN gateway support and comprehensive test case documentation. Most importantly, the company has expanded the number of tests to 36,000 cases that cover all the message specifications in RFC3261, RFC2543, RFC2327 and RFC2617 [19]. Unfortunately, the research team was unable to obtain a copy of the program for evaluation.

3.4 SIP Forum Test Framework (SFTF)

The SIP Forum Test Framework (SFTF) is an open source project hosted at sipfoundry.org. According to the developers, it was designed to test for common errors in devices in order to improve interoperability. SFTF provides both an easy way to write SIP device tests and a set of implemented test cases for typical errors made in SIP user agents. [23] The current version contains about 65 cases, which test for protocol implementation, authentication, registration, dialog/transaction processing, DNS, NAT capabilities and obsolete features [24].

3.4.1 Strengths

The framework allows a user to define new interoperability and vulnerability detection test cases via scripting. The included tests are specifically developed from commonly known implementation errors that cause problems.

3.4.2 Weaknesses

The SFTF "scope of tests" document enumerates each test case with call flow diagrams and sort descriptions. However, it does not include many pieces of information that would be useful to the user. First off, the listed source is either an individual's name or a reference number. The reference number does not correspond to RFC2543, the IETF's SIP torture test Internet draft or any document found on the SFTF website. Proper reference documentation is important because the test logic cannot be verified without extensive research or in-depth protocol knowledge. Secondly, the test descriptions provided in the SFTF documentation are generally not detailed enough to get a complete understanding of what is tested or the conditions that will cause a failure. The code and protocol analyzer outputs must both be examined to get the complete picture.

When testing against the user agent, several issues were identified. During the tests requiring registration, each case had to be run separately because the user agent did not have a function for initiating a REGISTER request. An unregister request by the UA, which is attempted automatically at program close, crashed the SFTF program. Thus, both the UA and SFTF had to be restarted after every test case. Furthermore, tests 303reg, 303inv and 208cseq crashed the SFTF program. The lack of robustness proved to be very frustrating because the features for running many tests at once (i.e. all the non-interactive tests) could no longer be used. Running each test by itself consumes far more time, especially since there is no function for specifying a range of tests to run. The SFTF program does not run tests in any discernable order, making it difficult to determine what tests have failed to run at the time of a program crash during a multi-function test. Several of the cases are designed to test functionality that is not standard, such as TCP connection handling. There is no place in the configuration to specify features implemented on the target and no message indicating that a test failure does not necessarily indicate a conformance problem.

3.4.3 Developer Claims and Analysis

The developers claim that the program does the following:

1. "[T]he SIP test framework…allows everyone with a little programming knowledge to write his own tests for SIP devices."
Analysis: The framework does make it much easier to write SIP interoperability tests than starting from scratch. Limited API documentation is available; however, it would be helpful if it also included a development guide for getting started.

2. "[A] bunch of implemented tests use [the] framework to test SIP user agents for typical known errors."
Analysis: There are only about 65 test cases, but they are highly focused on documented common errors affecting interoperability. Testing showed that the suite is capable of finding significant vulnerabilities.

In conclusion, SFTF provides a much more limited set of torture test cases than the PROTOS or SiVuS tools. However, it does have more tests for proper implementation of certain protocol specifications that can ensure interoperability. While the documentation is far from complete, it does give the tester some idea of the basic function of each case. As SFTF evolves, increased

robustness, more test cases and better documentation should make the program more useful to network engineers.

3.5 Commercial Security Appliances

Recently, several companies have released hardware appliance and software products designed to protect VoIP applications. Many of the firewall, intrusion detection/prevention and encryption security strategies developed for web technology can be transferred to VoIP systems with some application specific modifications. For example, firewalls are not normally able to pass SIP signaling packets, preventing external calls from entering a corporate network. However, firewalls designed for VoIP can reduce threats from malformed packets, voicemail attacks, spoofing, session high jacking and SIP spam (also known as SPIT) [25], while forwarding traffic for legitimate calls. Borderware, SecureLogix and TippingPoint have all brought network appliances to market within the last year that claim to significantly reduce VoIP security risk.

3.6 UnityOne

TippingPoint's UnityOne is a standard Intrusion Prevention System (IPS) that designates rule sets for VoIP vulnerabilities to assure voice quality of service. It can prevent denial of service attacks through threasholding and rate shaping [26]. The TippingPoint UnityOne 2400 unit can protect up to four network segments. A segment is protected when its traffic passes through a pair of ports on the IPS that are configured with filters and global settings. The device scans and reacts to network traffic according to the filter definitions. Each segment and device can use a different set of filters to manage and block traffic and malicious attacks. The Local Security Manager (LSM), a web-based management application, was used to administer the unit and manage reports during the test. Filters are used to detect attacks and abnormal protocol behavior.

3.6.1 Testing Method

The TippingPoint UnityOne was placed at the gateway of a network containing one X-Lite VoIP softphone. Packets were generated from an external PC-attached phone and sent through the UnityOne unit to the target softphone. An Ethereal packet analyzer running on the computer with the softphone was used to determine if the packets were blocked by the unit. Two tests were run to determine whether the UnityOne product was able to live up to developer claims. The first test had three parts—a TCP SYN flood, a UDP SYN flood and an ICMP (ping) flood. The second test involved using the PROTOS c07-SIP test suite to send SIP packets with malformed headers to test the effectiveness of the unit's filters at blocking malicious application layer data.

3.6.2 Strengths

The unit's specially designed hardware allows it to analyze packets at a very high rate. Weekly patches provide updates that ensure that the unit can protect against the latest threats. Because UnityOne was designed as a standard IPS rather than a VoIP device, it can also protect the rest of the network from threats such as standard DDoS attacks. The web-based GUI is easy to understand and use to configure the system.

3.6.3 Weaknesses

The TippingPoing UnityOne's high price tag (starting at around $25,000) will keep it out of reach from many smaller organizations. Because the unit was not designed as a VoIP gateway

like many competing commercial appliances, it is capable of protecting the system from a smaller set of VoIP-specific vulnerabilities. In fact, the system currently has only fourteen filters for SIP-based VoIP systems. Finally, testing revealed that the system does not block against all of the threats it claims. Specifically, a UDP flood at port 5060 was not blocked by the system (see Table 2 below).

Table 2: Flood test summary.

Threat	Result
TCP SYN Flood	Attack Blocked
UDP SYN Flood	Attack Not Blocked
ICMP Flood	Attack Blocked

Tests completed by forwarding malformed SIP packets generated by the PROTOS tool and covered by system filters also were not blocked (see Table 3 below). Additional test data can be found at http://zeuto.com/wiki/tiki-download_wiki_attachment.php?attId=7.

Table 3: Malformed header test summary.

Name	Test case number	The number of attempts / The number of detects
SIP-Method	#1	1400 / 1
SIP-Version	#255/# 498	870 / 0
SIP via Tag / SIP from Tag	#573/#823	450 / 0
SIP Call ID value	#1468	130 / 0
SIP Via Host Anomaly	#330/#436/ #452	87 / 0
SIP Via URL	#194	80 / 0
Content-Type Filed Anomaly	#2114	110 / 0
SIP From Field	#630, #832, #896	150 / 0

3.6.4 Developer Claims and Analysis

TippingPoint claims that the UnityOne:

1. "Can provide comprehensive attack prevention for VoIP against known and unknown zero-day cyber threats. UnityOne secures VoIP through Infrastructure Protection, VoIP Protocol Anomaly Protection and VoIP Application Protection. UnityOne assures VoIP Quality of Service and further protects against denial of service attacks through its patent pending thresholding and rate shaping capabilities [26]."

Analysis:
Our testing and analysis indicates that the TippingPoint UnityOne offers far from "comprehensive" attack prevention for VoIP networks. With our best efforts, we were unable to coax the system into blocking a significant number of malicious packets from even the most straightforward of methods. Even if the system did perform as promised, it still would cover

only a very small portion of known VoIP vulnerabilities. Finally, the architecture of the system as a normal IPS rather than a VoIP gateway prevents it from achieving the same level of protection as many competing commercial products.

When contacted regarding the results of our testing, TippingPoint technicians responded with "…our SIP filters are pretty exploit specific. True anomaly detection for VoIP won't be available until the 2.5 TOS release time frame." The statement directly conflicts with marketing material available from the company, which suggests that the system provides "comprehensive attack prevention" and specifically mentions "VoIP Protocol Anomaly Protection [26]." Interestingly, SC Magazine just named TippingPoint's IPS "Best Security Solution for 2005 [30]." The TippingPoint test results demonstrate how essential it is for companies to conduct testing of security equipment or hire an expert third party to do so before purchasing. As the test shows, company marketing materials and magazine reviews are not always accurate.

3.7 Other Commercial Appliances

Several other commercial appliance products are available that claim to mitigate security risk to VoIP systems. However, the research team was unable to obtain these additional devices for testing and analysis. Descriptions of the BorderWare SIPassure and the SecureLogix VoIP Firewall for Hybrid Networks follow to provide examples of other products with additional features that could provide better protection than the tested products. None of the following claims of the vendors have been verified.

3.7.1 SIPassure

According to BorderWare, the SIPassure appliance functions as a SIP proxy server, redirect server and registrar server. All SIP methods and RTP sessions are authenticated at the network border to prevent unauthorized packets from entering the network. A policy enforcement engine is dynamically configured to prevent against denial of service attacks. Rule sets determine if access from a specific IP or Uniform Resource Indicator (URI) should be restricted for a period of time after numerous abnormal activity is received (such as 10 calls within 30 seconds) from a user. The engine can also prevent malicious calling, VBombing and SPIT. A blacklist and whitelist allow for an override of dynamic enforcement. The risk of eavesdropping is reduced through signaling method authentication and encryption. These features, along with RTP traffic monitoring, also restrict call forking and redirection. SIP and RTP packet headers are analyzed for validity to prevent any malformed message attacks [27]. Another product that functions very similarly to SIPassure is the SIParator line from Ingate [28]. The main difference is that the SIParator lacks a method to prevent SPIT [25].

3.7.2 SecureLogix

The SecureLogix VoIP Firewall for Hybird networks sets itself apart by claming to provide security and management across VoIP and legacy circuit-switched networks. An extension of the ETM System firewall, the system adds wire-speed deep packet inspection to protect against application layer threats such as malformed packet attacks, signaling DoS and media DoS attacks. Call pattern anomaly detection can prevent SPIT, bandwidth abuse and toll fraud [29].

4 Findings

4.1.1 Vulnerability Scanner Tools Analysis

While robustness programs can be useful for unearthing poorly written programs, their limitations must also be understood. As explained by Dijkstra, "Program testing can show the presence of bugs, but never their absence [31]." Furthermore, the availability of testing tools may encourage some developers to rely on included test cases without developing their own. The conclusion that "it passed the test, it must be secure" is easy to reach when, in reality, no test program can test for an infinite number of cases. For instance, the c07-sip test cases only cover INVITE requests. Several products document that their test cases are based on the 2002 Internet Draft "Session Initiation Protocol Torture Test Messages." If all developers are using the same test cases, it could make the attacker's job easier by revealing what cases were not tested.

All of the vulnerability tools mentioned in this paper are still under development. Thus, the programs do not always perform as claimed. Most suffer from interface, robustness, scalability and functional issues. Rarely is the documentation adequate for a thorough analysis of the test results. In certain circumstances, it is difficult to determine the true effects of a test attack. For example, if a malformed packet does not crash a server, but causes it to hang for a second, the tester might not notice a problem with the server. However, if an attacker sent 1,000 of these packets to the server, a significant denial of service would occur.

4.1.2 Vulnerability Scanner Test Case Comparison

Table 4 below summarizes the features claimed by each vulnerability scanner developer. It provides a useful comparison of the potential for the tested and untested tools to detect security issues in VoIP implementations.

Table 4: Features claimed by vulnerability scanner tools

.

	Tested			Untested
Test Case Type	SIVuS	PROTOS	SFTF	Codenomicon
Malformed SIP Methods (robustness tests)	350+ INVITE, REGISTER, OPTIONS, ACK, CANCEL and BYE method checks	4,500+ INVITE method checks	25 INVITE, 1 OPTIONS method checks	36,000+ INVITE, REGISTER, OPTIONS, ACK, CANCEL and BYE method checks
TLS Support Check	1 check			
Authentication Verification	2 checks		7 checks	
Obsolete Feature Warnings			5 checks	
DNS Failure Recovery Verification			2 checks	
Dialog/Transaction Processing Conformance			20 checks	

4.2 Test Case Type Descriptions

Below are descriptions of the types of test cases used for comparison in table 1 above.

Malformed SIP Methods (robustness tests)
Robustness checks attempt to identify application layer programming flaws in the SIP implementation. Such errors can be exploited in a denial of service attack, which affects system availability (see section 2.3.4).

TLS Support Check
A check to verify whether Transport Layer Security (TLS) can be used to encrypt SIP signaling when run over TCP (also called SIPS). Encryption at the transport layer can be used to reduce the risk of many confidentiality and integrity related exploits (see sections 2.1.3 and 2.2.2).

Authentication Verification
Checks to verify that an implementation requires SIP messages to be authenticated. Authentication can mitigate many confidentiality and integrity related issues such as registration hijacking and session hijacking (see section 2.1.4).

Obsolete Feature Warnings
These alert the user if some part of the way the implementation handles SIP processing has been made obsolete. Obsolete functions can cause interoperability errors and open confidentiality, integrity or availability related vulnerabilities.

DNS Failure Recovery Verification
Test ensures that the implementation can recover in the case of a primary DNS failure to ensure availability.

Dialog/Transaction Processing Conformance
Non-standard dialog or transaction processing can cause interoperability errors and open confidentiality, integrity or availability related vulnerabilities.

4.3 Commercial Device Claims Comparison

Table 5 below summarizes the features claimed by each commercial product developer. It provides a useful model for comparing threats that could be mitigated through the devices.

Table 5: Attacks mitigated as claimed by company.

Threat	Untested		Tested
	BorderWare SIPassure	SecureLogix VoIP Firewall	TippingPoint UnityOne
DoS Attacks	X	X	X
Eavesdropping	X		
Call redirection	X		
SPIT	X	X	
Spoofing	X		
Malformed Message Attacks	X	X	X
RTP Session Hijacking	X		
RTP Injection	X		
Legacy network interaction vulnerabilities		X	

4.4 Mitigated Vulnerabilities

Table 6 below lists all of the vulnerabilities from section 2, and the tools from section 3 that claim to mitigate these risks. As can be seen, very few of the threats identified in this paper are addressed by any VoIP security tools available today.

Table 6: Tools that claim to mitigate vulnerabilities.

	Vulnerability	Tools Claim to Mitigate Risk
Data Link	Physical Attacks	
	ARP cache	
	ARP flood	
	MAC spoofing	
Internet	IP spoofing	SA
	Registration server, IP phone, MGCP, DNS, etc	SA
	Redirect via IP spoof	SA
	Malformed packets	
	IP frag	
	Jolt	
Transport	TCP / UDP flood	UO, SA, VF
	TCP / UDP replay	
Application	TFTP server insertion	
	DHCP server insertion (redirect)	
	DHCP IP address starvation	
	ICMP flood	
	SIP	
	Registration Hijacking	SA
	Call Hijacking (MGCP NotifiedEntity parameter)	
	Message body modification	
	RTP insertion	
	Spoof via header	SA
	Cancel / bye attack	
	Malformed method	SA, VF, OU, SI, PR, SF, CO
	Redirect method	SA
	RTP	
	SDP redirect	SA
	RTP payload	SA
	RTP message tampering	SA
	Encryption	SI
	Default settings / passwords	
	Disable unnecessary services HTTP, FTP, etc	
	Buffer overflow	SA, VF, OU, SI, PR, SF, CO
	SPIT	SA, VF
	Legacy Network Interaction	VF
	DNS Availability	SF

Tools Key

Code	Tested:		Code	Untested:
UO	TippingPoint UnityOne		CO	Codenomicon
SI	SiVUS		SA	Borderware SIPassure
SF	SFTF		VF	SecureLogix VoIP Firewall
PR	PROTOS c07-SIP			

5 Conclusion

In the past decade, the advancement of security related tools has greatly improved the network engineer's ability to assess security related risks and mitigate them across IP data networks. However, as can be seen from the analysis in the first section of this paper, VoIP still faces a number of security vulnerabilities. In fact, VoIP brings with it new security challenges that tool developers have only recently begun to address.

This paper shows that vulnerability scanners and active IPS/firewall appliances for VoIP are still very much in their infancy. Nearly every product found today was released within the last year. Interestingly, the fact that any security products are being released at all is somewhat remarkable in the historically reactive computer security industry. This suggests the importance and distinction of VoIP as an application. However, our testing has reveled that much work is needed on these tools if they are to become truly effective at mitigating significant threats.

Every vulnerability scanner tested suffered from at least one major weakness (such as usability or robustness) that would limit its effectiveness. Furthermore, the commercial appliance product which claimed to offer "comprehensive" protection was not even capable of blocking packets from the most straightforward of attacks. Because of the many discrepancies between developer product claims and actual risk mitigating abilities, network engineers should be careful when employing security tools. Whenever possible, the best strategy may be to test tools before trusting that they are effective. Hopefully, organizations such as the newly formed VoIP Security Alliance will one day be able to offer independent testing and certification of products [32].

One issue unveiled by this paper is that it is often quite difficult to determine exactly what kind of protection a device offers or what vulnerabilities a scanner searches for. Developers generally do a poor job of explaining their product's capabilities. For example, a device might claim to offer protection from a redirection attack. However, it is likely that the device is only able to protect a redirection from happening inside of the organization and not a call that terminates on the Internet. Thus, the user is still just as vulnerable to a hijacking attack. The limitations on getting information on exactly what devices are protecting against is likely done to prevent attackers from knowing enough about the system to attack it. Unfortunately, this policy also means that the customer could be receiving a largely ineffectual product without ever knowing it. The user may also be left in the dark about what additional measures must be taken to provide adequate security.

While there are clear benefits to employing effective security tools, there are some important things to keep in mind to regarding their use. Even as VoIP security tools evolve and improve over the coming years, they will never be able to detect all risks or block all attacks. There may be a tendency among naive engineers to plug in an appliance or run a security scan and declare the network secure. As we have demonstrated in this paper, if one was to do this with any of the tools listed here, many vulnerabilities would not be addressed in any way. While most prescribed "fixes" involve the encryption of data, many of the possible security breaches outlined in section one cannot be stopped through encryption alone. In real-time communications, encryption is often not an option due to latency issues. Only one undiscovered threat can result in the compromise of a system and few threats are addressed by tools today. Further research in the area of VoIP security is necessary to determine how more of the issues outlined in section 1 can be effectively mitigated in security tools.

In conclusion, none of the security tools evaluated were significantly effective for mitigating security risks in SIP-based VoIP networks. Almost all tools today are still under heavy development and will no doubt evolve as VoIP adoption increases. However, before VoIP security methods mature, significant attacks may pose some threats to confidence in the technology. Efforts such as the newly formed VoIP Security Alliance demonstrate that there is an interest to work proactively to solve VoIP security issues. In time, VoIP specific security tools should play an important role in securing systems.

References

[1] R. Mogull et al., "Predicts 2004: Critical Infrastructure Protection," *Gartner Research*, 14 Jan. 2005.

[2] Scottsdale, Ariz., "BorderWare Makes VoIP Safe," *BorderWare Press Release*, 14 Feb. 2005; http://biz.yahoo.com/prnews/050214/nym252_1.html

[3] B. Charney, "VoIP threats 'must be dealt with now,'" *CNET News.com*, 8 Feb. 2005; http://news.zdnet.co.uk/communications/0,39020336,39187096,00.htm

[4] J. E. Canavan, *Fundamentals of Network Security*, Boston: Artech House, 2001

[5] D.R. Kuhn, T. J. Walsh and S. Fries, "Special Publication 800-58: Security Considerations for Voice Over IP Systems," *NIST*, Jan. 2005; http://csrc.nist.gov/publications/nistpubs/800-58/SP800-58-final.pdf

[6] N. Dadoun, *Security Framework for IP Telephony*, tech. report TR-41.4.4, TR-41.4, Polycom, 15 Feb. 2002; http://ftp.tiaonline.org/tr-41/TR414/Public/2002-02-Vancouver/TR41.4.4/TR41.4.4-02-02-008SecurityFrameworknd.pdf

[7] J. Thalhammer, *Security in VoIP-Telephony Systems*, master's thesis, Graz Univ. of Technology, 2002.

[8] Si DF, Long Q, Han XH and Zou W, "Security mechanisms for SIP-based multimedia communication infrastructure," *IEEE Conf. on Comm*, Circuits and Systems (ICCCAS), ed. Proc. of 2nd ed., IEEE CS Press, 27-29 June 2004, pp.575-578.

[9] M. Thomas, "SIP Security Requirements," IETF Internet draft, work in progress, Nov. 2001.

[10] Yu-Sung Wu, S. Bagchi, S. Garg and N. Singh, "SCIDIVE:A Stateful and Cross Protocol Intrusion Detection Architecture for Voice-over-IP Environments," Conf. on Dependable Systems and Networks, Proc. of the 2004 Int'l Conf. on Dependable Systems and Networks (DSN'04), Jun.-July. 2004, pp. 433- 442.

[11] B. Bell, *VoIP Telephone Network Security Architectural Considerations*, tech. report TR-41.4.4-01-11-018, Cisco Systems, 6 Nov. 2001.

[12] S. Salsano, L. Veltri and D. Papalilo, "SIP Security Issues: The SIP Authentication Procedure and its Processing Load," *IEEE Network*, vol. 16, no. 6, Nov./Dec. 2002, pp.38- 44.

[13] L. McKeag, "How to cope with ARP attacks on LANs," Techworld, 20 July. 2004; http://www.techworld.com/security/features/index.cfm?featureid=727&Page=2&pagePos=3

[14] R. Spangler, "Packet Sniffing on Layer 2 Switched Local AreaNetworks," *Packetwatch Research*, Dec. 2003, pp.1; http://www.packetwatch.net/documents/papers/layer2sniffing.pdf

[15] C. Kaufman, R. Perlman and B. Sommerfeld, "DoS Protection for UDP- Based Protocols," Conf. on Computer and Comm. Security, Proc. of the 10th ACM Conf. on Computer and Comm. security, Washington DC, 2003, pp. 2-7.

[16] "Advanced Networking Management Lab (ANML) Distributed Denial of Service Attacks(DDoS) Resources," *Pervasive technology labs*, Indiana Univ., 2001; http://www.anml.iu.edu/ddos/types.html

[17] "Intrusion Prevention: The Future of VoIP Security", Tech whitepaper, Tipping Point, 2004.

[18] I. Dubrawsky, "Safe Layer 2 Security In-Depth," *Cisco*, 2004, pp.12 http://www.cisco.com/warp/public/cc/so/cuso/epso/sqfr/sfblu_wp.pdf

[19] "Codenomicon SIP Test Tool," Codenomicon, 2005**;** http://www.codenomicon.com/testtools/sip/sip-tech.html

[20] C. Weiser, M. Laakso and H. Schulzrinne, *Security Testing of SIP Implementations*, tech. report, Univ. of Oulu and Columbia Univ, 2003; http://www1.cs.columbia.edu/~library/TR-repository/reports/reports-2003/cucs-024-03.pdf

[21] *SiVuS User Guide*, VoIP Security Forum, 2004; http://vopsecurity.org/index.php?module=dpDocs&func=display&mid=4

[22] "PROTOS Test-Suite: c07-SIP," tech. report, Oulu Univ. Secure Programming Group, Univ. of Oulu, 2003; http://www.ee.oulu.fi/research/ouspg/protos/testing/c07/sip/

[23] "SIP Forum Test Framework (SFTF) – a testing software for SIP", SIPFoundry, 2004; http://www.sipfoundry.org/sftf/index.html

[24] C. Jennings et al., "Scope of Tests", IETF Internet draft, work in progress, SIP, 2004; http://www.sipforum.org/documents/test_cases_draft_05_sftf.pdf

[25] W. Rash, "BorderWare Firewall Fights VoIP Threats," the Channel Insider, 14 Sep. 2004; http://www.thechannelinsider.com/article2/0,1759,1646055,00.asp

[26] "Voice Over Internet Protocol (VoIP) Security," Tipping Point; http://www.tippingpoint.com/solutions_voip.html

[27] "SIPAssure SIP Firewall," Borderware, 2004; http://www.borderware.com/pdfs/sipassure.pdf

[28] "Ingate SIParators," Ingate Systems; http://www.ingate.com/siparators.php

[29] "VoIP Application Firewall & QoS Tools Highlight ETM(R) SystemVersion 5.0 From SecureLogix(R),” Yahoo Finance, 7 Feb. 2005; http://biz.yahoo.com/prnews/050207/dam010_1.html

[30] L. Craddick, "TippingPoint's Intrusion Prevention System Awarded Best Security Solution 2005 by SC Magazine," *TippingPoint Press Release*, 2005; http://www.tippingpoint.com/pdf/press/2005/SCAward_022205.pdf.

[31] E. Dijkstra, "Notes on Structured Programming," On the Reliability of Mechanisms, 1970; http://www.cs.utexas.edu/users/EWD/ewd02xx/EWD249.PDF

[32] "Voice over IP Security Alliance (VOIPSA)," VoIP Security Alliance, 2005; http://voipsa.org/index.html

Appendix A: SiVuS Test Result Summary

The following information provides a summary of the most important findings from running all test cases against the Asterisk PBX. Full test data can be found at: http://zeuto.com/wiki/tiki-index.php?page=SiVusRawData.

281 of 360 tests were reported as “high” risk.

Of the test results, 2 inaccurate results were found in the report:

```
Check 13000
Report Description:  Verifies the ability of the UA to authenticate REGISTER
requests.
Report Recommendation: It appears that the target UA does not authenticate
REGISTER requests using UDP. This configuration allows unathorized [sic]  users
to generate malicious messages in order to hijack user registration and
ultimately divert calls or perform other undesired functions. Change the
configuration of the SIP component to require authentication of REGISTER
requests.
Analysis:  The description and first part of the recommendation from the report
is incorrect.  Tests using different UA's and single message generation clearly
shows that Asterisk does indeed have the ability to authenticate REGISTER
requests.  The second part of the recommendation is correct, in that the server
does allow unauthenticated connections.   The test was repeated a second time,
following the recommendation of the program to require authentication.  The
test passed.
Check 12000.1
Report Description: This check verifies the ability of the UA to authenticate
INVITE requests.
Report Recommendation: It appears that the target UA does not authenticate
INVITE requests using UDP. This configuration allows unathorized [sic]  users
to generate malicious messages. Change the configuration of the SIP component
to require authentication of INVITE requests.
```

Analysis: The report result is inaccurate, as this output clearly shows that the Asterisk server did indeed require authentication for an INVITE request:

```
192.168.0.32  ->  192.168.0.15  SIP/SDP  Request:  INVITE  sip:1002@192.168.0.15,
with session description
192.168.0.15 -> 192.168.0.32 SIP Status: 407 Proxy Authentication Required
```

Appendix B: PROTOS c07-SIP Test Suite Result Summary

This appendix provides a short summary of the tests after all the c07-SIP test cases were run with the PROTOS tool against both the Asterisk PBX and the LinPhone softphone. Ethereal protocol analyzer captures from these tests are available at http://zeuto.com/wiki/tiki-index.php?page=ProtosRawData.

All test cases against the Asterisk PBX server passed the test criteria (see section 3.3). Tests on the LinPhone soft phone caused the program and Gnome window manager to stop responding on test cases 2, 3 and 4. The test was ended here as the program was already shown to have many serious flaws.

The PROTOS tool was found to crash with the following error:

```
java.lang.RuntimeException: Internal error, invalid test case file
```

on test cases after 16 when tested with J2SE 1.4.2 on both Debian Linux and Fedora Core 3. All tests ran properly with J2SE 1.4.2 on Windows XP.

Appendix C: SFTF Test Suite Result Summary

This appendix provides a short summary of the results from running the full SFTF test suite against the SJ Phone, LinPhone and X-Lite user agents. Full test data can be found at http://zeuto.com/wiki/tiki-index.php?page=SftfRawData. Descriptions of the test cases can be found at http://www.sipforum.org/documents/test_cases_draft_05_sftf.pdf.

Case #	Description	SJPhone Result	LinPhone Result	X-Lite Result
301	Digest Authentication of REGISTER without qop	PASS	PASS	PASS
302inv	Digest Authentication of INVITE without qop	Timeout	FAIL (wrong auth reply)	PASS
302bye	Digest Authentication of BYE without qop	Timeout (unable to agree on media codecs)	FAIL (wrong auth reply)	PASS
303reg	Digest Authentication of REGISTER with qop	SFTF Crash	PASS	PASS
303inv	Digest Authentication of INVITE with qop	SFTF Crash	FAIL (wrong auth reply)	PASS
304	Authentication retry timer	PASS	PASS	PASS
202	Valid characters protocol torture	FAIL (no reply)	FAIL (no reply)	FAIL (no reply)
212	R-URI unknown scheme	FAIL (no reply)	FAIL (no reply)	FAIL (no reply)
501	Correct via returning	PASS	PASS	INITIALIZED*
216	Unacceptable request offering	FAIL (wrong reply type)	FAIL (wrong reply type)	FAIL (wrong reply type)
226	Multipart body	PASS	WARNING (rejected request)	PASS
801c	Rport support	PASS	PASS	WARNING (rport returned without value)
201	Valid torture request	PASS	INITIALIZED*	FAIL (no reply)
205	Trailing octets in UDP	PASS	INITIALIZED*	PASS
211to	Missing header required fields	FAIL (missing reply)	UA CRASHED	FAIL (INVITE accepted)
401	Server-driven re-registration period	FAIL	FAIL	FAIL
501	Correct via returning	PASS	PASS	INITIALIZED*
905	Request merging	Timeout	PASS (forked INVITE rejected) WARNING (the second forked INVITE should be rejected with 482, instead received '842') ERROR (timeout)	Timeout
511	Correct route set instruction	PASS	SFTF Crash	PASS
208cseq	Regular scalar fields with overlarge values	SFTF Crash	FAIL (no reply)	SFTF Crash
225reuse	TCP Handling	FAIL (doesn't use TCP)	Fail (doesn't use TCP)	Fail (doesn't use TCP)
211callid	Respond with 400 if Call-ID missing	FAIL (no reply)	CRASHED UA	FAIL (no reply)
207	Respond with 400 if content length longer than message	FAIL (no reply)	FAIL (no reply)	FAIL (INVITE accepted)
213	Respond with 420 if unknown "Require" element	FAIL (no reply)	FAIL (wrong reply type)	FAIL (INVITE accepted)
228	UTF in display names accepted	PASS	WARNING (rejected with 404)	PASS
503c	No record-route in negative replies	PASS	PASS	FAIL
215	Max forward = 0 should be processed	PASS	FAIL	PASS
209	Return a 400 on undetermined quoted	FAIL (no reply)	Fail (no reply)	FAIL (no reply)

	string in display			
225close	TCP handling close	FAIL (doesn't use TCP)	Fail (doesn't use TCP)	Fail (doesn't use TCP)
211cseq	Respond with 400 if cseq missing	FAIL (no reply)	CRASHED UA	FAIL (INVITE accepted)
210	Invalid request-URI	FAIL (accepted INVITE)	FAIL (wrong reply)	FAIL (no reply)
204	Long values in header fields	FAIL (no reply)	INITIALIZED*	PASS
214	Unknown content type	FAIL (accepted INVITE)	FAIL (wrong reply)	FAIL (accepted INVITE)
224	OPTIONS support	FAIL (request rejected)	FAIL (no reply)	PASS
601can	Proper generation of CANCEL	PASS	PASS	PASS
206	Unusual reason phrase	INITIALIZED*	PASS	PASS
901	Presence of RFC3261 branchID	PASS	PASS	PASS
603c	Correct loose routing	FAIL (no reply)	FAIL (no reply)	PASS

* The result description "INITIALIZED" is not mentioned in the documentation. It is unclear to the testers what this means, why it is present in the tests results, or what the proper response to it should be.