Darshak Dobariya, Jagdish Gajjar / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 3, May-Jun 2012, pp.2666-2675

Threats In SIP Based VoIP Systems

Darshak Dobariya, Professor Jagdish Gajjar.
(Student, M.tech , Electrical Department, Veermata Jijabai Technological Institute, Mumbai)
(Professor , Electrical Department, Veermata Jijabai Technological Institute, Mumbai)

ABSTRACT
Security tools such as protocol analyzers, vulnerability assessment utilities and security monitoring utilities are amongthe common tools in a security professional's arsenal. Such tools have reached a high level of dependence among securityprofessionals for evaluating potential vulnerabilities in such areas as operating systems, device configuration, networking protocols and applications. However, these tools have their limitations, such as (1) where they are applied, (2) how they areimplemented and (3) how they are maintained and updated. Furthermore, while such tools are fairly robust for more maturetechnology, it remains difficult to develop comprehensive security tools for emerging technology. Voice over Internet Protocol is an example of such an emerging technology. This paper explores the known VoIP-related vulnerabilities andtests several of the more popular open source and commercial VoIP security tools with the intention of demonstrating the gap that exists between vulnerability and detection. Understanding this gap will help to identify what issues need to be addressed in the future development of VoIP security tools.

Keywords - VoIP, Security, Vulnerability, Tool, SIP

I. INTRODUCTION

On the Internet, popular applications and devices tend to become popular targets for 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 likely target because of its growing popularity. [1] Furthermore, VoIP presents new challenges in that it differs from traditional voice (i.e., circuit-switched telephony) in a number of ways. For example, no single entity controls the development and monitoring of VoIP. Implementers (and end users) may be empowered to configure these systems as they see fit. This leads to the issue of possible mis configuration, 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 few wide spread attacks unique to VoIP systems, the potential exists. The majority of the public will soon rely on VoIP and wide spread attacks could be devastating and significantly impact commerce and public safety. People have become accustomed to the high availability of 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 address VoIP. However, it is not clear that that these tools operate as thoroughly as may be required. Of course, all tools have their limitations; however, a tool should be able to perform the task that the developers claim. Furthermore, together these tools should provide a reasonable coverage of the potential vulnerabilities.

This paper examines the functionality of current VoIP security tools to determine their limitations. As this paper will demonstrate, there are a significant number of vulnerabilities that the tools failed to detect. The paper also demonstrates the gap that exists between the known vulnerabilities and the coverage provided by the tools.Additionally, this paper describes the potential problems that might arise with the installation and use of these tools, which could lead to other problems, including misconfiguration and misinterpretation of the data.

II. Vulnerabilities

The vulnerabilities in VoIP encompass not only the flaws inherent within the VoIP application itself, but also 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 classic areas of information security: confidentiality, integrity, and availability (CIA). For purposes of organization, we have separated these vulnerabilities based on the layers of the TCP/IP networking model (i.e.,network interface layer,

internet layer, transport layer, and application layer), although we recognize that many
vulnerabilities cross layers. Several aspects of network security have been omitted from this list of vulnerabilities because they are outside the relevance of paper. For example, non repudiation, access, and accounting have been left out of the vulnerabilities section despite their fundamental importance of network security. 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 many 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. [5] Furthermore, security measures such as authentication, address filtering, and alarms for when devices are disconnected can mitigate the risks involved in physical security. In a separate paper, we have identified and described the vulnerabilities impacting or relating to VoIP. (See http://spot.colorado.edu/~sicker/VoIPTools.htm). In identifying these vulnerabilities we undertook substantial secondary research (of which, [6-23] represent a small part of this literature review) and cross-tabulated these
findings with information from CERT as well as from several major software vendors. In this literature review, we found that are a substantial number of VoIP vulnerabilities and that there is considerable effort underway to identify and address these known vulnerabilities. However, even with this effort, it appears that the vulnerabilities are still very much beyond the scope of the tools presently available to security professionals. Below in table 1, we provide a reduced description of the vulnerabilities identified in the aforementioned paper. 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 the areas of network security they affect. This chart will be used in later sections to evaluate the comprehensiveness of the VoIP tools tested.

Table 1.

Layer	**Attack Vector**	**Conf.**	**Integrity**	**Availa.**
Network Interface	Physical Attacks	x		x
	ARP cache	x	x	x
	ARP flood			x
	MAC spoofing	x	x	x
Internet	IP spoofing			
	Device	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 DHCP			x

	starvation			
	ICMP flood			x
	SIP			
	Registration Hijacking	x	x	x
	MGCP Hijack	x	x	x
	Message 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 tampering	x	x	x
	Encryption	x	x	x
	Default configuration	x	x	x
	Unnecessary services	x	x	x
	Buffer overflow	x	x	x
	Legacy Network	x	x	x
	DNS Availability			x

III 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 exploitations can be expected to grow as the number of new systems grows. [3, 6] As VoIP systems become ever more prevalent and risk grows, security professionals need to make sure they are taking the proper precautions to prevent security breaches. However, little work has been undertaken to evaluate the usefulness of these security tools. 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. 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. 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 exceptional elements are not handled correctly by the user agent or server, it can cause the program to hang, crash or even provide a means for gaining unauthorized access. Thus, it is important for software engineers to run thorough conformance and robustness checks against VoIP software. Tests can also be useful to the network engineer to verify robustness of the product they want to deploy. SIP implementations should not be assumed to be 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 [23].

III.I 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 meet developer claims of functionality. Since all products have different functionality, a direct comparison on all grounds was not feasible. Rather, an attempt was made to evaluate the strengths and weaknesses of each tool at a more general level. We make no pretense that we will be able to evaluate all VoIP security tools. Rather, we sought to examine the more commonly accessible tools.

III.II 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, 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.

III.II.I Strengths

The SiVuS scanner's main strength is that it was designed with a Windows GUI, which makes it more user friendly than several of the other vulnerability scanners tested. Reports are generated in an easy to read web format. While this report does not contain all of the information necessary for proper evaluation, 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.

III.II.II Weaknesses

One of the greatest weaknesses of SiVuS is the lack of information to properly analyze the test output. 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 it 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 real vulnerabilities. For instance, the program could be looking for a certain response packet to each malformed packet within a timeout period. If the server doesn't respond because it is set not to respond to 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 widely used, it is all the more critical that the documentation be updated so users can at least easily 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 while using the program that required a program restart. For example, saved configurations and the file name in "log all scanning activity" are only loaded when the program is initialized. Also, 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 (for additional information on these tests, see http://spot.colorado.edu/~sicker/VoIPTools.htm). Fourth, running the test cases repeatedly fails to find a target at first, 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.

III.II.III Developer Claims and Analysis

In the user guide, the developers claim the following capabilities: 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 performed 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 aforementioned web site for details).
3. "Authentication of registration requests." Analysis: The claim was verified through testing, however an error was found in the report (see aforementioned web site for details).
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 for a security professional. However, the program represents a significant effort to develop a vulnerability scanner that will hopefully continue to improve.

III.III 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, which inject exceptional elements into SIP INVITE messages, including SDP. Monitoring of the SIP implementation the program is run against 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."

III.III.I Strengths

One of the great strengths of this program is its simple design. There are 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 a good understanding of the capabilities and functionality of the program. All of the test cases are outlined in a table so that there is little question what every case does. The source code is available as a reference if more in-depth knowledge is needed. Finally, the test cases were designed using a comprehensible methodology.

III.III.II 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 presented compatibility problems, even though it was supposedly written 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 can be a time consuming process of limiting the test cases and repeating until only the single case that caused the crash is run.

III.III.III. 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 [23]."
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." [23] Codeomicon Inc. has developed the PROTOS tool into a commercial test tool to include 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 (codeomicon.com). Unfortunately, the research team was unable to obtain a copy of the program for evaluation.

III.IV 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. [16] The current version contains about 65 cases, which test for protocol implementation, authentication, registration, dialog/transaction processing, DNS, NAT capabilities and obsolete features [17].

III.IV.1 Strengths

The framework allows for scripting of new interoperability and vulnerability detection test cases. The included tests are specifically developed from common known implementation errors that cause problems.

III.IV.II 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. There ference number does not correspond to the SIP RFC, the IETF's SIP torture test Internet draft or any document found on the SFTF website. More useful references are important because now 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 either what is tested or the conditions that will cause a failure. The code and protocol analyzer output must both be examined to get the complete picture. When testing against the user agent, several issues that would be considered frustrating to the user 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 noninteractive 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 multiple test functions do 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. 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.

III.IV.III 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 there was also some kind of "how to" 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 effecting 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 implementation of certain protocol specifications, which can be important for ensuring 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.

III.IV.IV Open Source Solutions

It is easy to discount the above analysis because of its focus on open source products and the known limitations (in terms of support and upgrades) of such software. However, we found that VoIP security user groups commonly discussed these products as useful tools for assessing VoIP installs. While we do agree that these tools are indeed useful, their limitations must also be realized.

III.V Commercial Security Appliances

Recently several companies have released appliances designed to test VoIP security. We have tested several of the commercial products; however, due to various concerns, we are reluctant to publish the results in a way that identifies specific vendors. We are willing to

state that none of the products we tested provided a complete solution and that each had limitations in terms of vulnerability detection, user interface or installation procedures. Furthermore, in section 4.3 we compare the claims of the various commercial and non-commercial products against a list of known vulnerabilities. As the reader will see, even the commercial products only address a small part of the vulnerability space.

IV. Findings

In this section, we present our findings. This includes a discussion of the strength of the vulnerability scanning tools, the tests applied to these tools and their ability to mitigate general vulnerabilities

.

IV.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." 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 for. All of the vulnerability tools mentioned in this paper are still under development. Thus, the programs don't always perform as claimed. Most suffer from interface, robustness and functional issues. Rarely is the documentation adequate for a through 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 doesn't 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.

IV.II Test Case Type Descriptions

Below are descriptions of the types of test cases used for comparison in table 2. The intention of the list is to provide a collapsed description of the types of features offered by the tools.

1.) *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.
2.) *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.
3.) *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.
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.
5.) *DNS Failure Recovery Verification:* Test ensures that the implementation can recover in the case of a primary DNS failure to ensure availability.
6.) *Dialog/Transaction Processing Conformance:* Nonstandard dialog or transaction processing can cause interoperability errors and open confidentiality, integrity or availability related vulnerabilities.

IV.II.1.Vulnerability Scanner Test Case Comparison

Table 2 below summarizes the features claimed by each vulnerability scanner developer. The chart categorizes the vulnerabilities and shows the number of 'checks' within those categories for each of the tested tools. It provides a useful comparison of the potential for the tested tools to detect security issues in VoIP implementations. As can be seen, the coverage across broad categories is weak.

Table 2: Features claimed by vulnerability scanner tools.

Test Case Type	**SiVuS**	**PROTOS**	**SFTF**
1. Malformed SIP Methods (robustness tests)	1. Malformed SIP Methods (robustness tests)	4,500+ INVITE method checks	25 INVITE, 1 OPTIONS method checks

2. TLS Support Check	1 check		
3. Authentication Verification	2 checks		
4. Obsolete Feature Warnings			5 checks
5. DNS Failure Recovery Verification			2 checks
6. Dialog/ Transaction Processing Conformance			20 checks

IV.IIIMitigated Vulnerabilities

Table 3 lists all of the vulnerabilities from section 2, and the tools from section 3 that claim to mitigate these risks. Also included in this list are a number of well known security test tools. Some of these tools we were able to test in our labs others we were only able to judge based on the claims made be the developers. As can be seen, the VoIP security tools tested address a limited set of the identified vulnerabilities. And together (if combined into a suite of tools), they address less than half of the known vulnerabilities. Furthermore, as we have shown in section 3, the tools do not always perform as claimed, so these results are generous at best. The findings suggest that security professionals should be cautious in trusting the network of their networks to such
tools.

Table 3: Tools that claim to mitigate vulnerabilities

Layer	**Attack Vector**	**Tools**
Network Interface	Physical Attacks	
	ARP cache	
	ARP flood	
	MAC spoofing	
	`	
Internet	IP spoofing	SA
	Device	SA
	Redirect via IP spoof	SA
	Malformed packets	
	IP frag	
	Jolt	
Transport	TCP / UDP flood	XX, SA,VF
	TCP / UDP replay	
Application	TFTP server insertion	
	DHCP server insertion	
	DHCP starvation	
	ICMP flood	
	SIP	
	Registration Hijacking	SA
	MGCP Hijack	
	Message modification	
	RTP insertion	
	Spoof via header	SA
Layer	Attack Vector	Tools
	Cancel / bye attack	
	Malformed method	ALL

	Redirect method	SA
	RTP	
	SDP redirect	SA
	RTP payload	SA
	RTP tampering	SA
	Encryption	SI
	Default configuration	
	Unnecessary services	
	Buffer overflow	ALL
	SPIT	SA, VF
	Legacy Network	VF
	DNS Availability	SF
	Tool Key	
Code		**Tested**
XX		A commercial product
SI		SiVuS
SF		SFTF
PR		Protos c07-SIP
		Untested
CO		Codenomicon
SA		Borderware
VF		SecureLogix

V. Conclusion

In this paper, we have demonstrated that many of the popular VoIP security tools (1) do not cover the extent of the known vulnerabilities, (2) do not always provide the coverage the developers claim and (3) may be difficult to install and properly configure. Clearly, VoIP security tools are still in their infancy and continue to evolve. Efforts like the VoIP Security Alliance [4] show a new commitment to the advancement of VoIP security research and software. However, it is very difficult to develop tools that can address the various vulnerabilities outlined in this paper. We are not suggesting that these tools are not useful, nor are we saying that they should not be used. Indeed, such tools can be a very useful in identifying specific vulnerabilities; however, as is the case with any security tool, it is important to realize the limitations of the tool and not allow their use to create a false sense of security.

References

[1] R. Mogull, C. Moore, D.L. Fraley, et. al, "Predicts 2004:Critical Infrastructure Protection," Gartner Research, January14, 2005.

[2] "BorderWare Makes VoIP Safe," BorderWare PressR e l e a s e , [o n l i n e] . Availble:http://biz.yahoo.com/prnews/050214/nym252_1.html,.

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

[4] "Voice over IP Security Alliance (VOIPSA)" [online].Availble: http://voipsa.org/index.html,

[5] N. Dadoun, "Security Framework for IP Telephony",Polycom White Paper. 15 Feb. 2002

[6] D. Richard Kuhn, Thomas J. Walsh, Steffen Fries,"Special Publication 800-58: Security Considerations for VoiceOver IP Systems", National Institute of Standards and Technology, Jan 2005

[7] Johann Thalhammer, "Security in VoIP-Telephony Systems", Masters Thesis, Graz University of Technology,Austria, 2002

[8] Si DF, Long Q, Han XH, Zou W, " Security Mechanisms for SIP-Based Multimedia Communication Infrastructure." Proceedings of 2nd IEEE Conference on Communications, IEEE Press, 2004.

[9] M. Thomas, "SIP Security Requirements", IETF Internet-D r a f t R e t r i e v e d f r o m http://www.softarmor.com/wgdb/docs/draft-thomas-sip-sec-req-00.txt, 20 Feb. 2005.

[10] E. Dijkstra. "Notes on Structured Programming," On the Reliability of Mechanisms, 1970. Retrieved from

http://www.cs.utexas.edu/users/EWD/ewd02xx/EWD249.PDF, 20 Feb. 2005.

[11] S. Salsano, L. Veltri, D. Papalilo, "SIP Security Issues: The SIP Authentication Procedure and its Processing Load", Network, IEEE, Vol. 16, Iss. 6, Nov/Dec 2002, p.38- 44

[12] Charlie Kaufman, Radia Perlman, Bill Sommerfeld,"DoS Protections for UDP-Based Protocols", Conference on Computer and Communications Security, Proceedings of the 10th ACM conference on Computer and communications security, Washington DC, 2003, p. 2-7

[13] Yu-Sung Wu, Saurabh Bagchi, Sachin Garg, Navjot Singh, "SCIDIVE: A Stateful and Cross Protocol Intrusion Detection Architecture for Voice-over-IP Environments", Conference on Dependable Systems and Networks, Proceedings of the 2004 International Conference on Dependable Systems and Networks (DSN'04), 28 June-1 July 2004, p. 433 - 442

[14] "VoIP Telephone Network Security Architectural Considerations", Cisco Systems, 6 Nov 2001

[15] "Intrusion Prevention: The Future of VoIP Security",Tipping Point, 2004

[16] "SIP Forum Test Framework – a test foundry for SIP". SIPFoundry, http://www.sipfoundry.org/sftf/index.html

[17] "Scope of Tests", SIP Forum, http://www.sipforum.org/documents/test_cases_draft_05_sftf.pd

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

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

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

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

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

[23] C. Weiser, M. Laakso, H. Schulzrinne, "Security Testing of SIP Implementations", 20 Feb, 2005 http://www1.cs.columbia.edu/~library/TRrepository/ reports/reports-2003/cucs-024-03.pdf