Evaluating DoS Attacks Against SIP-Based VoIP Systems

M. Zubair Rafique, M. Ali Akbar and Muddassar Farooq
Next Generation Intelligent Networks Research Center (nexGIN RC)
FAST National University of Computer & Emerging Sciences (NUCES)
Islamabad, Pakistan
Email: {zubair.rafique, ali.akbar, muddassar.farooq}@nexginrc.org

***Abstract*—The multimedia communication is rapidly converging towards Voice over Internet – commonly known as Voice over Internet Protocol (VoIP). Session Initiation Protocol (SIP) is the standard used for session signaling in VoIP. Crafty attackers can launch a number of Denial of Service (DoS) attacks on a SIP based VoIP infrastructure that can severely compromise its reliability. In contrast, little work is done to analyze the robustness and reliability of SIP severs under DoS attacks. In this paper, we show that the robustness and reliability of generic SIP servers is inadequate than commonly perceived. We have done our study using a customized analysis tool that has the ability to synthesize and launch different types of attacks. We have integrated the tool in a real SIP test bed environment to measure the performance of SIP servers. Our measurements show that a standard SIP server can be easily overloaded by sending simple call requests. We define the performance metrics to measure the effects of flooding attacks on real time services - VoIP in SIP environment – and show the results on different SIP server implementations. Our results also provide insight into resources' usage by SIP servers under flooding attacks. Moreover, we show that how a well known open source SIP server can be crashed through 'INVITE of Death' - a malformed SIP packet maliciously crafted by our tool.**

I. Introduction

The global communication market is experiencing a rapid increase in demand for novel Internet based multimedia applications. Session Initiation Protocol (SIP) is an application layer signaling protocol used for establishing, controlling and annihilating the media sessions of these applications. The common services of SIP in multimedia application include network gaming, interactive TV, Voice over IP (VoIP), PC clients, multiparty conferencing, video on demand, presence and instant messaging [1].

A recent market survey indicates that VoIP accounts for 49.7% of total voice traffic at the end of year 2007 [2]. With this level of penetration of Internet telephony, SIP servers are becoming a hot pursuit for imposters and intruders. A recent survey by SANS Institute supports our argument by suggesting that SIP servers are among the SANS top 20 security risks [3]. Another study shows that VoIP servers are among the top 5 emerging cyber security threats for the year 2009 [4]. However, robustness of SIP servers against different types of threats is not well-studied or understood. We, therefore, undertake an empirical study to evaluate the performance and robustness of different implementations of SIP proxy servers under DoS attacks.

The purpose of our study is to help both VoIP vendors and academia better understand different vulnerabilities in the existing SIP servers and how adequately they are protected against them. The mitigation of DoS attack is not within the scope of this paper. We believe robustness analysis of SIP servers is an important step towards designing DoS protection strategies for SIP servers. In our study, we specifically try to investigate a number of relevant issues:

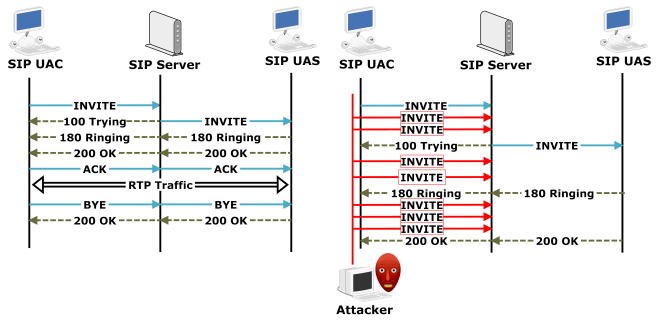
- What is the impact of simple DoS attacks on the performance of SIP proxy servers?
- Are existing well-known SIP servers robust against emerging threats?
- What is the 'breaking point'[1] of a SIP server after which a complete DoS occurs?
- How realistic is the threat of malformed packets attack?

In order to systematically conduct our investigative study, we have developed a tool that can launch two types of attacks: (1) flooding of call requests at different rates to launch DoS, and (2) generating malformed packets that contain mutation of strings, crafted on specific positions, to exploit vulnerabilities in parsing or implementation codes. Our tool uses SIPp – an open source SIP traffic generator – for generating call requests [5]. The tool automatically generates performance reports that provide two types of performance metrics: (1) SIP related metrics help us in understanding the impact of performance degradation on VoIP calls, and (2) system related metrics that are useful for analyzing the impact of attacks on the hardware resources like CPU usage.

We have selected four well-known SIP proxy servers for our robustness study. We use transactional stateful proxying mode [6] of the servers. We deploy one server at a time in our real test bed and then launch, using our tool, different types of attacks on it. Using the report generation module of our tool, we analyze the performance degradation of each server that provides the understanding of behavior of a given SIP server under DoS attacks.

The rest of the paper is organized as follows. Section II briefly introduces the threat model used to study the robustness of SIP server, the different performance metrics that we define to do a comparative study of the robustness of a SIP server and

[1]We define *breaking point* as an attack scenario in which only 50% of the requested calls are completed.



(a) Benign Call Model (b) INVITE Flood

Fig. 1. SIP Call Models

discussion on our attack synthesis and analysis tool. In Section III, we discuss the real testbed we have deployed for our experiments. We discuss and analyze results of our robustness study in Section IV. In Section V, we briefly describe the related work in the field of vulnerability analysis of servers. Finally, we conclude our paper with an outlook to our future research.

II. Security Evaluation, Metrics and Attack Synthesis/Analysis Tool

In this section, we first describe the security threats related to SIP servers. Then, we define metrics for measuring the robustness of the SIP servers under these security threats. Afterwards, we describe our tool that is capable of launching these attacks and calculating the performance of the SIP servers in terms of these metrics.

A. SIP Security Threats

The easiest way to launch Denial of Service (DoS) attacks on a SIP proxy server is to flood it with a large number of unwanted call requests. As a result, its resources – internal memory buffers, CPU and bandwidth – are exhausted and it is unable to provide service even to the legitimate users (see Figure 1). The requirements of resources is dependent on the fact whether SIP server is configured for stateless or stateful mode and is using authentication or not [7].

Moreover, SIP is also prone to malformed message attack in which attackers generate non-standard SIP messages, that are intelligently crafted to exploit vulnerabilities in the SIP parser or in poor implementation of a SIP server. An imposter can, using a malformed packet, overflow the specific string buffers, add large number of token characters and modify fields in an illegal fashion. As a result, a server is tricked to reach an undefined state, which can lead to call processing delays, an unauthorized access and a complete denial of service. We also show how an intelligently crafted single malformed message can crash a server. We call it *Invite of Death*.

In our experiments the intensity of flood attacks varies from 1000 `INVITE` packets/sec to 10000 `INVITE` packets/sec. Figure 1 shows the SIP call models of benign and `INVITE` flood scenarios. The parsing attacks are launched with the help of malformed SIP packets generated by our tool.

B. Performance Metrics

The motivation of identifying and selecting relevant performance metrics is very important for our study because: (1) they help us in examining that how different SIP servers behave under DoS attacks, (2) they help academia to better understand the severity of DoS in SIP environment and a VoIP vendor to do risk analysis of his/her business operations, and (3) they provide information from an end user perspective about the quality of service that they should expect from a SIP server in case of DoS attacks. We, therefore, propose two types of metrics: (1) SIP based metrics, and (2) SIP independent system metrics. SIP based metrics define the quality of service from an end user perspective. If these metrics are degraded in DoS attacks, it would mean service unavailability to the end users. In real world scenario it would result in VoIP customers dissatisfaction that would indirectly lead to loss of revenue and creditability of the vendor. On the other hand, if system metrics are degraded, it can lead to a complete denial of service which of course poses a significant threat.

We define the SIP based metrics as:

Call Completion Ratio (CCR). The ratio of the number of benign calls[2] that are successfully completed during an attack scenario to the number of calls successfully completed in no-attack scenario.

Call Establishment Latency (CEL). The average delay that a SIP client experiences between dialing of a number and successfully establishing the call. Specifically it is the average delay between sending of an `INVITE` request message by a SIP client and receiving of corresponding `200 OK` response from the SIP server (see Figure 1).

Call Rejection Ratio (CRR). The ratio of the number of benign calls rejected by a SIP server during an attack scenario to the number of calls rejected in normal no-attack scenario. The metric determines the effective *loss* of potential resources of a SIP server under attack scenarios. It also represents the fraction of SIP clients unable to get services from the server.

Number of Retransmitted Requests (NRR). The number of request messages which are retransmitted due to server timeout or network congestion. The metric models the congestion level in a network because of large number of `INVITE` packets. If NRR increases significantly during an attack, CCR, CEL and CRR will also degrade.

We now define SIP independent system metrics that show us whether a machine, hosting a SIP server, is able to meet requirements of SIP clients. In extreme attack scenarios, a machine might become unresponsive or the operating system might crash. These metrics are:

CPU usage. The average CPU usage of the machine hosting a SIP server.

CPU interrupts rate. The rate at which different interrupts are received by the CPU.

[2]We use the term *benign calls* to represent the calls requested by legitimate users.

Interrupt handling time. The average time taken by the CPU to service these interrupts.

C. Attack Synthesis and Analysis Tool

Our attack synthesis and analysis tool[3] consists of three important modules: (1) client configuration module, (2) attack generation module, and (3) report generation module. The client configuration module configures the SIP clients to generate normal call load on a SIP server. The clients call each other randomly through the SIP server. Once connected, they start variable length voice sessions consisting of RTP traffic.

1) Client Configuration Module: The reasons to generate a normal call flow on the server are twofold: (1) to create a real-world normal call scenario for a SIP server, (2) to systematically analyze the degradation in performance – experienced by legitimate users – under attack scenarios. The simulated clients generate calls randomly with an average load of 3000 calls per min. The tool configures the SIP client instances on separate machines with the parameters like call rate, media port, time out value, remote host parameters and IP address.

2) Attack Generation Module: The attack generation module in our tool can launch flooding or malformed packet attacks. The tool is capable of generating 9600 malformed packets of various categories: Null mutations, Space mutation, Utf-8 invalid character mutation, Escape characters mutation, Token string mutation and ASCII characters mutation. Mutations are carried on every possible position in the SIP header. Similarly our module launches DoS attack by flooding large number of unwanted `INVITE` messages to SIP server.

3) Report Generation Module: The job of the report generation module is to gather statistics from the SIP clients and the server. Each SIP client generates a report in the `csv` file format during the experiment. Similarly on the server machine the statistics of hardware resources are also calculated and logged in realtime. Once an experiment finishes, report generation module communicates with SIP clients, SUT, the attacking node, and generates the report of above-mentioned performance metrics.

III. Experimental Testbed

Now, we describe our experimental testbed that we have used to evaluate the performance of different SIP servers under different types of attack scenarios. Figure 2 shows the architecture of testbed used in our experiments. Note that the clients – the caller and the called parties – are instantiated on separate machines to make accurate measurements of performance metrics. The User Agent Clients (UACs) initiate the calls; while the User Agent Servers (UASs) are the SIP clients that receive the call from UACs and start the dialogue. All SIP related traffic is proxied through the SIP servers while the RTP traffic is routed directly between the UASs and UACs and has no impact on System Under Test (SUT). The flooding and parsing attacks are separately launched from an attacker node. To conclude, our testbed consist of five components: (1) SIP User Agent Clients (UACs), (2) SIP User Agent Servers (UASs), (3) Imposter nodes, (4) SIP server, and (5) Analysis Machine.

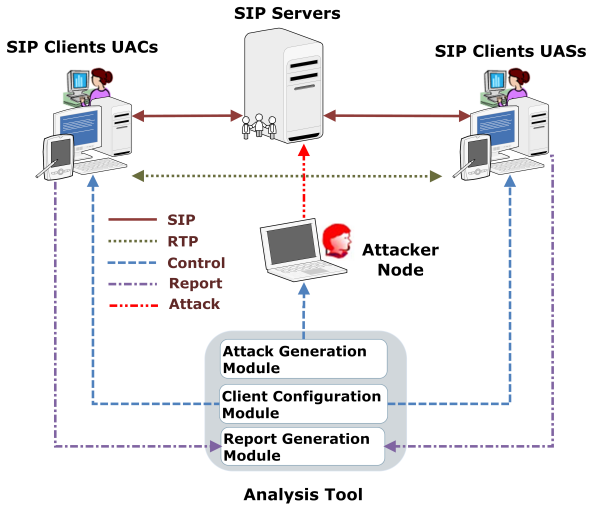


Fig. 2. Experimental Testbed

TABLE I
SIP Servers Used in Experiment

Server Name	Version	License	O.S
OpenSER	1.1.1	GNU GPL	Linux
PartySIP	2.2.2	GNU GPL	Linux
OpenSBC	1.1.5	MPL,GPL,LGPL	Linux
MjServer	1.6	GNU GPL	Linux

A. SIP UACs and UASs

The benign users of the system are simulated as SIP UACs (callers) and UASs (callees). These clients are implemented using a modified version of SIPp. We have adapted SIPp in order to make it inter operable with our custom analysis tool.

B. Attacker Nodes

We have implemented attacker nodes as *bots* running on separate machines. These bots are implemented in `C++` and are capable of launching DoS attacks on a SIP server[4]. We can configure and manage them from our analysis tool.

C. SIP Proxy Servers

We have selected four SIP servers for our robustness study. The criteria for selection of servers are: high performance, wide deployment and public availability. Table I shows the list of servers [8], [9], [10], [11], their version, license type and the operating system on which they are deployed. The selected servers are well-known and offer basic features of SIP: proxying, registration and redirection. The servers are compliant with the IETF's standard SIP specifications [6]. We configure stateful proxying without authentication in our experiments because this is the commonly used configuration by VoIP service providers. This configuration helps them in doing

[3]http://ims-bisf.nexginrc.org/SIPTool.php/

[4]Currently, these bots can launch only SIP flood and parsing attacks.

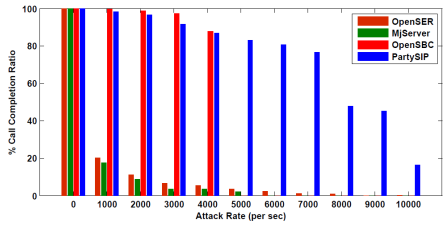


Fig. 3. Percentage CCR vs. Attack Rate

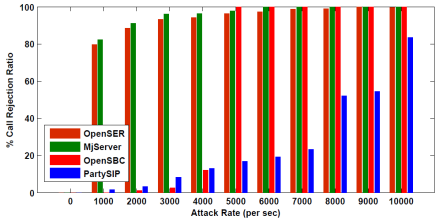


Fig. 5. Percentage CRR vs. Attack Rate

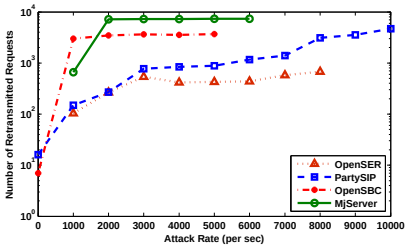


Fig. 4. NRR vs. Attack Rate

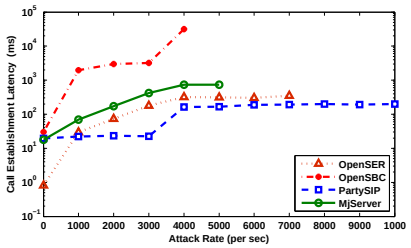


Fig. 6. CEL vs. Attack Rate

accounting and billing on a per call basis. Other proxying scenarios are simple extensions of the stateful scenario [6].

D. Attack Synthesis and Analysis Machine

This machine is a high-end system which hosts our attack synthesis and analysis tool. It controls the UACs, UASs and the attacker node. It also monitors the state of the SIP servers and calculates performance metrics for the servers.

E. Hardware Specifications

The testbed has been designed in our IMS laboratory with an isolated network. The experiments are done on an isolated network to factor out unwanted background traffic and minimize interference from other processes. The machines hosting UACs, UASs, the SIP servers (SUT), and the intruder nodes are Intel(R) 2.20 GHz processors with 2 GB RAM and 240 GB disk drives. All machines are running Linux Fedora Core 8 with a kernel version 2.6.23.1. The machines are connected to a 3 COM Gigabit switch.

IV. ROBUSTNESS RESULT OF GENERIC SIP SERVERS

We now present and analyze the robustness of different SIP servers in terms of CCR, CRR and NRR metrics under different attack scenarios. The results of our experiments show that DoS attacks significantly degrade these metrics which can undermine the ability of a VoIP service operator to continue its operations. We expect CCR to decrease, CRR to increase and consequently NRR would also increase with an increase in the intensity of the attack rate. Degradation of these performance metrics is a direct measure of the threat level on VoIP infrastructure of an operator by an imposter. Consequently, the operator can do risk analysis of the potential loss in revenue at this threat level.

We have shown different performance metrics under attack, obtained from our analysis tool, in Figures 3, 4, 5, 6, 7, 8 and 9. Now we discuss the impact of DoS attacks on each metric.

A. Analysis of CCR, CEL, CRR and NRR

In Figure 3, we plot CCR (in percentage) of four servers for different flooding rates of `INVITE` packets. (Note that the bar graph of different servers follows the same order as shown in legend.) Recall that CCR is a measure of successfully established calls under attack scenarios. In contrast, CRR in Figure 5 shows the benign calls that are rejected during an attack scenario. An attack rate of **zero** on x-axes of all figures shows normal operating scenarios without any attack. It is interesting to note that an attack rate of just 1000 `INVITE` packets/sec reduces CCR of OpenSER and MjServer to less than 20% (or 80% CRR as shown in Figure 5). This level of degradation in performance in effect is a successful DoS attack. Similarly in Figure 4 we see that retransmission requests jump to 100 and 1000 for OpenSER and MjServer respectively once CCR reduces (note that Y-axis is on the logarithmic scale). This is because 80% of customers receive the busy signal and hence redial the number. Another important observation in Figure 3 is that OpenSBC is robust to DoS attacks till 4000 `INVITE` packets/sec; while PartySIP can counter 8000 `INVITE` packets/sec. Note that NRR of OpenSER is less compared with that of PartySIP even though OpenSER has high CRR and low CCR compared with PartySIP. We investigated the issue and found out that OpenSER replied with the server busy (5xx response) once it observed high load. Our simulated client simply rejects the call once it receives 5xx response message from the server.

From Figure 6, we see that the call establishment latency initially rises exponentially and then achieves a steady state value. If we compare it with Figures 3 and 5, we conclude that the ratio of established calls almost reaches to zero under

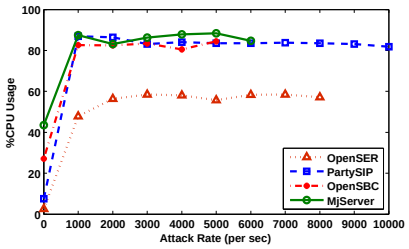


Fig. 7. Average CPU usage vs. Attack Rate

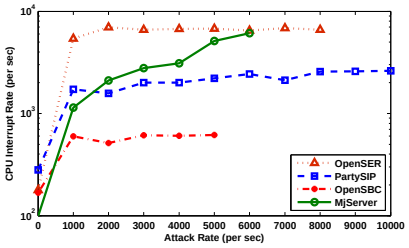


Fig. 8. CPU Interrupt Rate vs. Attack Rate

high attacks rates, hence only a very small fraction of calls are established. But the established calls have acceptable latency. Since CEL is measured on successfully established calls only; therefore, MjServer, OpenSBC and OpenSER have less data points in Figure 6 compared with PartySIP.

To conclude, it is obvious from Figures 3, 4, 5 and 6 that none of the servers is capable of defending itself against DoS attacks. But some servers like PartySIP are definitely better designed and robust against high attack rates. An imposter needs to select an appropriate rate for `INVITE` flood attacks to successfully launch a DoS attack on a given SIP server.

B. CPU usage, Interrupt Rate and Interrupt Handling Time

We now discuss the impact of system parameters under attacks. We see in Figure 7 that the CPU usage increases exponentially with an increase in the attack rate. Just for an attack rate of 1000 `INVITE` packets/sec, the usage increases from 1%, 3%, 23% and 40% for OpenSER, PartySIP, OpenSBC and MjServer respectively to 40%, 80%, 81% and 82% respectively. This further confirms that servers start suffering from DoS attack because CPU is kept busy processing malicious call requests that leaves little time for processing legitimate calls. We further investigated the reasons behind such a dramatic increase in the CPU usage. An important outcome is that even at an attack rate of 1000 `INVITE` packets/sec, the number of hardware interrupts exponentially increases from few hundreds to few thousands (see Figure 8). As a result, the servers are spending most of their time in processing the interrupts. This also corresponds to our findings in Figure 7. It is interesting to note that the interrupt service time of MjServer rises exponentially with an increase in the attack rate. While other three servers show graceful degradation in service time till the point they completely crash. We note that OpenSER has the lowest CPU usage even though it has largest number of interrupts. This is explained by the fact

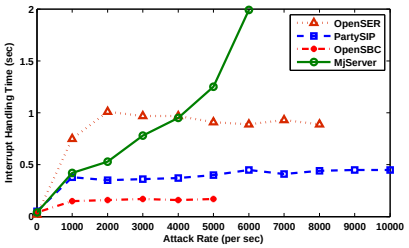


Fig. 9. Interrupt Handling Time vs. Attack Rate

that it refuses to accept more call requests above a predefined threshold value. Therefore, the total number of active call sessions remains within a predetermined limit. Afterwards OpenSER simply sends server busy response to all requests. We argue that Figures 7, 8 and 9 provide valuable insights that how these metrics can be utilized to design and develop an intelligent host based intrusion detection system for SIP.

C. Parsing Attack Result

Another important contribution of our work is the parsing attack generation module, which has the capability to automatically create malformed messages. The malformed messages exploit vulnerabilities in the SIP parser, as a result, a server can be crashed by a single packet. We call it as *Invite of Death*. We generated 9600 intelligently crafted malformed packets. We have successfully crashed OpenSBC server on a number of occasions with our malformed packets. One of the example packets that crashed OpenSBC server is shown in Figure 10. Note that overflow of colons at the end of "Via" field represents a malformed syntax in the packet. The vulnerability highlights the weak parsing technique in the implementation of OpenSBC. We have already reported the vulnerabilities to the development team of OpenSBC[5]. They have now redesigned their parser in which the parsing vulnerabilities are removed and new version is released as 1.1.5-82. This shows the utility of our attack generation module.

In comparison we have not been able to crash OpenSER, PartySIP and MjServer. The reason being the best coding practices adopted during their implementation. For example in OpenSER several parsing operations – using UTF-8 encoding – take constant time during parsing of the string because it also takes as an input the length of a string along with the string. Furthermore it "uses lazy parsing to only parse those headers necessary rather than naively parsing all of them. Last but not least, it incrementally parses only needed fields within a header" [7]. MjServer generates certain type of string exceptions on receiving the malformed packets; however, we are unable to crash it.

V. Related Work

The reliable performance of SIP server is critical under DoS attacks. To the best of our knowledge, no empirical study is done to analyze robustness of SIP servers against

[5]The advisory for this vulnerability is available at [12].

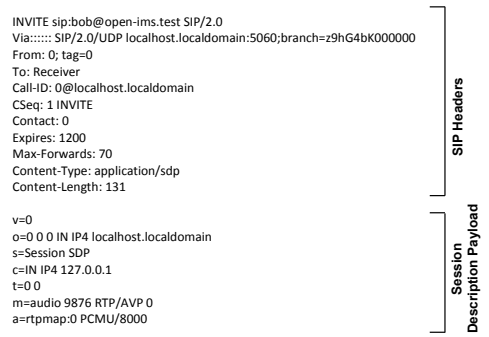


Fig. 10. Malformed Packet (crashes OpenSBC)

DoS attacks. As a result, very little is known or understood about robustness of SIP servers. The authors of [13] have empirically evaluated the SMTP servers against DoS attacks. An important work is reported in [14] in which the authors conceptually discussed the impact of different types of attacks on VoIP infrastructure. They have conceptually identified exploitable server resources, such as memory, CPU usage and bandwidth and presented abstract guidelines to ensure SIP servers' robustness under different attack scenarios. But they paid no attention to empirically analyze the performance hit of SIP servers under attack. The authors of [15] summarize the features of vulnerability analysis tools available for VoIP. Similarly, the authors of [16] suggest using a Virtual Private Network (VPN) solution to circumvent attacks on a SIP server. The authors did not discuss how their scheme is resilient against DoS attacks. The authors of [7] have experimentally evaluated the SIP proxy (OpenSER), using micro-benchmarks, and analyzed the performance of OpenSER as a function of selecting different configuration modes of the server. They have also ignored robustness analysis of SIP servers against different types of DoS attacks. The authors of [17] proposed a stateful method of detecting flooding attacks against SIP servers. While the authors of [18] proposed a two layer DoS pervention architecture that handles both SIP flooding and malformed packet attacks on a standard VoIP network.

VI. Conclusion and Future Work

In this paper, we have evaluated robustness of four well-known SIP servers against Denial of Service (DoS) attacks. We defined metrics to quantify robustness of a SIP server under DoS attacks. We have also developed a SIP vulnerability analysis tool and integrated it into our real test bed. We have done experiments on it to answer four questions raised in Section I: (1) What is the impact of simple DoS attacks on the performance of SIP proxy servers?, (2) Do existing well-known SIP servers are robust against emerging threats?, (3) What is the 'breaking point' of a SIP server after which a complete DoS occurs? and (4) How realistic is the threat of malformed packets attack? We have summarized results in Table II. The important conclusion is that existing well-known SIP servers can be knocked out of service by launching simple `INVITE` flood attacks. However, servers like PartySIP and OpenSBC are more robust (see Table II) compared with

TABLE II
Breaking Point and Vulnerability of SIP Servers

SIP Servers	Breaking Point	Parsing Vulnerability
MjServer	200 INVITE/sec	No
OpenSER	500 INVITE/sec	No
OpenSBC	4000 INVITE/sec	Yes
PartySIP	8000 INVITE/sec	No

OpenSER and MjServer. Note that the breaking points of MjServer, OpenSER, PartySIP and OpenSBC are 200, 500, 4000, 8000 `INVITE` packets/sec respectively. We have also shown that how an intelligently crafted malformed packet crashed OpenSBC. This emphasizes the need for developing robust SIP parser by following secure coding standards. The practice is followed in OpenSER, PartySIP and MjServer.

Our study justifies the need to implement efficient realtime SIP Intrusion Detection System (IDS), which can protect a SIP VoIP infrastructure from emerging SIP threats. In future, we plan to extend our robustness analysis on SIP severs by testing them in presence of specialized SIP based IDS/IPS and by providing the insights about the architecture, coding techniques and design of different SIP servers which inherently make them resilient to DoS attacks.

Acknowledgments

This work is supported by the National ICT R&D Fund, Ministry of Information Technology, Government of Pakistan. The information, data, comments, and views detailed herein may not necessarily reflect the endorsements of views of the National ICT R&D Fund.

References

[1] C. Pavlovski, "Service Delivery Platforms in Practice," *Comm. Mag., IEEE*, vol. 45, no. 3, pp. 114–121, 2007.
[2] The-VoIP-Network, "VoIP Market Trends," 2008, http://www.the-voip-network.com/voipmarket.html.
[3] SANS-Institute, "SANS Top-20 2007 Security Risks," 2007, http://www.sans.org/top20/.
[4] G. T. I. S. C. (GTISC), "Emerging Cyber Threats Report for 2009," 2008, http://www.gtiscsecuritysummit.com/pdf/CyberThreatsReport2009.pdf.
[5] R. Gayraud *et al.*, "SIPp," http://sipp.sourceforge.net.
[6] J. Rosenberg *et al.*, "SIP: Session Initiation Protocol," *RFC 3261, IETF, 2002*, 2002.
[7] E. Nahum *et al.*, "Evaluating SIP Proxy Server Performance," in *Proc. NOSSDAV '07*, 2007.
[8] Open-source Session Border Controller (OpenSBC) , http://www.opensipstack.org.
[9] PartySIP, SIP Proxy Server, http://www.partysip.org.
[10] MjServer, SIP Proxy Server, http://www.mjsip.org/mjserver.html.
[11] O. S. E. R. (OpenSER/kamailio), http://www.kamailio.org.
[12] nexGIN RC, "OpenSBC Remote Denial of Service Vulnerability," 2009, http://ims-bisf.nexginrc.org/OpenSBC-vul.html.
[13] B. Bencsath *et al.*, "Empirical Analysis of Denial of Service Attack Against SMTP Servers," in *CTS 07*, 2007, pp. 72–79.
[14] D. Sisalem *et al.*, "Denial of Service Attacks Targeting a SIP VoIP Infrastructure: Attack Scenarios and Prevention Mechanisms," *IEEE NET.*, vol. 20, no. 5, p. 26, 2006.
[15] S. McGann *et al.*, "An Analysis of Security Threats and Tools in SIP-Based VoIP Systems," in *Second VoIP Security Workshop*, 2005.
[16] V. Miroslav *et al.*, "Performance Comparison of Secure and Insecure VoIP environments," *Proc. KTTO '08*, 2008.
[17] E. Chen, "Detecting DoS attacks on SIP systems," in *1st IEEE Workshop on VoIP Management and Security, 2006*, 2006, pp. 53–58.
[18] S. Ehlert, G. Zhang, D. Geneiatakis, G. Kambourakis, T. Dagiuklas, J. Markl, and D. Sisalem, "Two layer Denial of Service prevention on SIP VoIP infrastructures," *Computer Communications*, 2008.