SIP Vulnerability Scan Framework

Mitra Alidoosti*
Department of Computer Engineering, Iran University of Science and Technology,Tehran. Iran
Alidoosti@comp.iust.ac.ir
Hassan Asgharian
Department of Computer Engineering, Iran University of Science and TechnologyTehran. Iran
Asgharian@iust.ac.ir
Ahmad Akbari
Department of Computer Engineering, Iran University of Science and Technology,Tehran. Iran
Akbari@iust.ac.ir

Received: 29/Jul/2013 Revised: 14/Jun/2014 Accepted: 22/Jul/2014

Abstract
The purpose of this paper is to provide a framework for detecting vulnerabilities in SIP (Session Initiation Protocol) networks. We focused our studies on the detection of SIP DoS related vulnerabilities in VoIP infrastructures because of their generalization. We try to find weaknesses in SIP enabled entities that an attacker by exploiting them is able to attack the system and affect it. This framework is provided by the concept of penetration testing and is designed to be flexible and extensible, and has the capability to customize for other similar session based protocols. To satisfy the above objectives, the framework is designed with five main modules for discovery, information modeling, operation, evaluation and report. After setting up a test-bed as a typical VoIP system to show the validity of the proposed framework, this system has been implemented as a SIP vulnerability scanner. We also defined appropriate metrics for gathering the performance statistics of SIP components. Our test-bed is deployed by open-source applications and used for validation and also evaluation of the proposed framework. The main contributions of this paper are its non-destructive manner in identifying vulnerabilities and incorporating the penetration testing ideas and steps in the overall architecture of our framework. We also defined appropriate metrics that help us to identify vulnerabilities in a black box penetration testing.

Keywords: Vulnerability Scanner; SIP; Denial of Service Attacks; Framework; Evaluation.

1. Introduction

Voice over IP protocols (VoIP) simply enables two devices to transmit and receive real-time audio traffic that allows their respective users to communicate. VoIP architectures are generally partitioned into two main groups: signaling and media [1]. Signaling covers both abstract notions, such as endpoint naming and addressing, and concrete protocol functions such as parameter negotiation, access control, billing, proxying (routing), and NAT traversal [2]. The media transfer aspect of VoIP systems generally includes a simpler protocol for encapsulating data, with support for multiple codecs and content security. A commonly used media transfer protocol is RTP. There exists an RTP profile that supports encryption and integrity protection (SRTP), but it is not yet widely used. The RTP protocol family also includes RTCP, which is used to control certain RTP parameters between communicating endpoints. In spite of the media transport layer of VoIP infrastructures, its signaling layer can accept different signaling like H.323, Skinny and SIP. In this paper we focus on the SIP which is the most widely used protocol in the standard VoIP architectures and next generation networks [3, 4]. Unfortunately, because of the interoperability requirements with the existing telephony infrastructure, its new features, and the speed of development and deployment, VoIP protocols and products contain numerous vulnerabilities that have been exploited. Most of these vulnerabilities are the result of the complexity of VoIP systems which demonstrates itself both in terms of configuration options and implementation issues. As a result, VoIP systems represent a very large attack surface [1]. So it is expected that security problems arising from design flaws (e.g. exploitable protocol weaknesses), undesirable feature interactions (e.g. the combination of components that make new attacks possible), unforeseen dependencies (e.g. compromise paths through seemingly unrelated protocols), weak configurations, and many other implementation flaws.

Vulnerability scanning is the process of assessing a variety of vulnerabilities across information systems (including computers, network systems, operating systems, and software applications) and allowing early detection and handling of known security problems [5]. A vulnerability scanner can help to identify rogue machines, which might endanger overall system and network security, helps to verify the inventory of all devices on the network [5]. The inventory includes the device type, operating system version and patch level, hardware configurations and other relevant system information. This information is useful in security management and tracking. There are general tools for vulnerability assessment and scanning of some application layer protocols but because of the special

* Corresponding Author

vulnerabilities of VoIP architectures, there is no well-known and widely acceptable tool in this field.

Therefore we have proposed a SIP vulnerability scanner framework for evaluating VoIP components against well-known SIP attacks. We focused our studies on the detection of SIP DoS related vulnerabilities in VoIP infrastructures because of their generalization. Although our proposed solution is general and has no assumption about the underlying VoIP component (i.e. user agent devices and proxy servers) but because of our previous experiences, we focused on SIP proxy servers and present its results on our experimental test-bed. The main contributions of this paper are its non-destructive manner in identifying vulnerabilities and incorporating the penetration testing ideas and steps in the overall architecture of our framework. We also defined appropriate metrics that help us to identify vulnerabilities in a black box penetration testing.

The other parts of this paper are organized as follows: The next section reviews the literature and some related works. Our proposed solution is expressed in section 3 and our experimental setup and evaluation of the presented system is defined in section 4. Finally the conclusion is abstracted in section 5.

2. Literature Reviews and Related Works

One of the main approaches to security assessment of computer networks and systems is penetration testing. Penetration testing tools perform a non-destructive attack to check the security status of an organization network and distinguish its vulnerabilities. Generally it has three steps which are done sequentially: discovery, attack and report.

The process of penetration testing contains system analysis to identify the potential vulnerabilities of systems which arises because of misconfigurations or implementation faults. The penetration testing process is categorized in to two broad groups [6]: black box and white box. In black box testing, it is assumed that there is no knowledge about considered network. We selected this approach in our proposed SIP vulnerability scanner. Thus, in discovery step all required information about the given target is collected. This information contains SIP enabled devices and their footprints. Since other steps depend tightly on this step, the pen-tester must take suitable time to complete this phase. In attack step, non-destructive attacks are imposed on the target and according to their effects, being inferred that the target is vulnerable or not. In taking report step, the sufficient report is prepared to notify the organization about available vulnerabilities.

Reference [7] demonstrates VoIP specific security assessment framework to perform automated VoIP specific penetration tests. This framework searches and detects existing vulnerabilities or misconfigured devices and services. This security assessment tool mentions DoS attacks, but flooding attacks are not considered, so could not verify how the behavior of SIP systems may change under system load during flooding attacks. This framework architecture contains three main modules that perform the required tasks such as discovering as much as possible information from the devices in the network, storing and providing all collected information in a usable format and finally launches penetration tests and perform attack actions using gathered information. Other related work for VoIP penetration testing is [8]. It measures the vulnerability of SIP-based VoIP systems during security attacks. It considers some categories of DoS attacks and defines the availability of the system under test (SUT) for its validation that we used its main idea for evaluation in our paper.

In [8], the main focus of availability is on the user interaction during attack times. The ratio of successful call rate during attack's period to pre-attack times is measured; furthermore the re-transmission number of each call is calculated that represents the influence of attacks. Since the main focus of [8] is on attack generation, it is likely to damage the SUT. Therefore we try to solve this problem by considering the potential vulnerabilities of SUT (using the result of discovery phase) and plan the non-destructive attacks based on that.

Reference [9] presents a security management framework for VoIP. In order to estimate the SIP and RTP related security vulnerabilities and threats of VoIP; a fuzzy packet tool is developed. The functionality of the proposed framework defines in XML scenarios. Depends on the physical location of this tool, different tasks can be performed such as man in the middle attacks, user enumeration and password testing for a registration server, ARP injection in order to intercept network traffic or just protocol level fuzzing.

We inspired the steps of our framework from the phases of penetration testing. In other words, we assess the existence of vulnerabilities on a given target by discovering the proper target and plan a proper operation against it for realization. Reference [7] demonstrates an acceptable framework for vulnerability assessment; we get the generality of our framework from it that we customized it based on our knowledge about the VoIP attacks. Reference [8] presents a good idea for measuring the effects of attacks, it divides the duration of the test in 3 parts: pre-attack phase, attack phase and post-attack phase. The perception of this fact that one target is vulnerable to a specific attack or not, is measured by changes of the system behavior to normal users. We use this idea in our framework to figure out the sensitivity of considered platform to attacks. We define three criteria and measure them during pre-attack, attack and post-attack stages. The changes of these criteria during these phases simply detect that our target is vulnerable or not. The details of our proposed framework are defined in the next section. Because of comprehensiveness and importance of DoS flooding attacks in SIP, our main focus in this paper is on them. SIP flooding attacks are reviewed in many papers like [10,11] that we don't review them here.

3. Proposed Approach to SIP Vulnerability Assessment

As it is said in the previous sections, our main goal is to design a system for vulnerability scanning of VoIP systems. The main output of the proposed approach is an evaluation tool for comparing different implementations of SIP components in handling known attacks. We focused on the vulnerabilities that led to denial of service attacks. Since it does not affect the generality of our problem, we limited our studies on SIP proxy servers and three broad classes of SIP DoS attacks but the proposed solution is general and can be extended to any VoIP components and VoIP related vulnerabilities. In other words, we aim to mention whether the given SIP proxy is vulnerable against specific DoS attacks or not. In fact we want to explore the weaknesses of given sip proxy which the attacker misuses them and intrudes the sip proxy.

Our main contribution is to proposed SIP vulnerability scan architecture. The architecture of our proposed solution is shown in Figure 1. It has five main modules that are as follows:

1. Discovery Module
2. Information Model Module
3. Operation Module
4. Evaluation Module
5. Report Module

In the following subsections, the detail design of these modules is described.

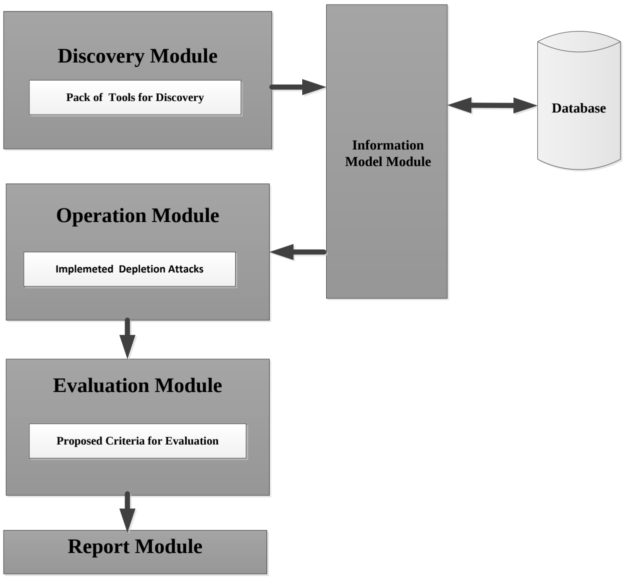

Fig 1. Architecture of the proposed framework

3.1 Discovery Module

The main objective of this module is to recognize the active hosts. Discovery Module is shown in Figure 2.

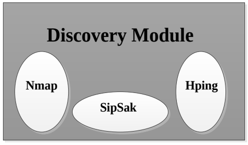

Fig. 2 Discovery Module

This module contains specific tools for fingerprinting the given network. It also provides useful information about the active host such as IP address, open ports, running services, MAC address and also some other information about the specific node in the VoIP network. This module works automatically and finds all active hosts with SIP enabled services. The last step of this phase is recognizing the type of the active host such as being the proxy server or the user agent client. In this module, the Nmap [12] tool is used to identify active hosts in a VoIP network, we configured the SIPSak [13] to discover the type of host and Hping [14] is used to diagnose active and inactive hosts.

3.2 Information Model Module

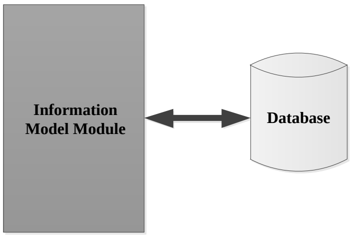

Fig. 3 Information Model Module

The Information Model Module is shown in Figure 3. This module uses some online repositories like NVD [15] to find out the related vulnerabilities of the enumerated hosts by previous module. In this stage by using stored information in the database, potential vulnerabilities within the entity will be discovered. According to recognized vulnerabilities, appropriate attacks are chosen. Type of entity and selected attacks are saved in the database and this information is given to the next module. Therefore this module is responsible for the following two functions:

1. Selecting types of applied attacks for the system under test.
2. Updating the database according to newly discovered vulnerabilities.

In other words, according to information obtained from the previous step and the type of entity, the type of applied attack is selected and the database is updated based on the type of entity and selected attacks.

3.3 Operation Module

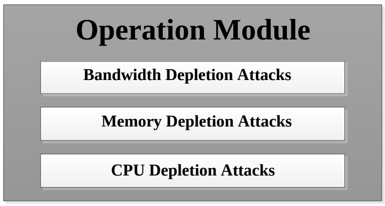

Fig. 4 Operation Module

The responsibility of operation module is to apply selected attacks against the target. This module and evaluation module will start to run simultaneously. By analyzing the underlying traffic through the target, the evaluation unit can determine whether the target is vulnerable to corresponding attack or not. So in this module the selected attacks are applied to the target and the vulnerable target can be identified by evaluation module. The applied attacks in this module are bandwidth depletion attacks, memory depletion attacks and CPU depletion attacks. The architecture of Operation Module is shown in Figure 4.

Bandwidth depletion attacks by creating a large number of redundant messages try to occupy the bandwidth. The aim of memory depletion attacks is consuming SIP entity's memory so that it is not able to respond to legitimate demands of users. For generating memory attack we produce messages that extend call setup time, so the target is forced to hold call information in its memory until the call is finished, therefore the memory will be occupied longer than usual and the sessions prolong which cause to cease the memory. By producing a certain number of these kind of messages, memory of the target will be occupied and will not have sufficient space for legitimate users. CPU depletion attacks are generated by creating messages that need additional processing to keep the processor busy. So that the target does not have enough time to process messages receiving from other legal entities and users. These kind of attacks usually make by using malformed messages or authentication based attacks.

3.4 Evaluation Module

As stated before, evaluation and operation module start simultaneously. This module assesses passing traffic through the SIP entity, so the vulnerability of the target will be extracted. For detecting vulnerabilities of SIP entity a new metrics are defined. By measuring these criteria, the vulnerability of the target against applied attacks can be diagnosed.

3.4.1 Proposed Scheme to Identify Vulnerabilities

As shown in Figure 5, simulation period was considered as T3 seconds. During the entire simulation period, normal traffic is available between the proxy server and other existing users.

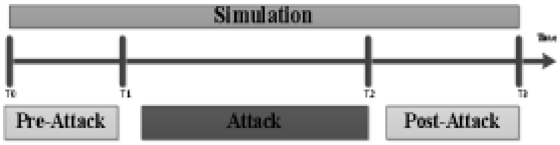

Fig.5 proposed scheme to identify vulnerable hosts

The attack is applied in the time interval [T1, T2]. The defined criteria are measured during time interval [T1, T2] and [T0, T1], [T2, T3]. Time intervals [T0, T1] and [T2, T3] indicate non-attack period and the period [T1, T2] indicates attack period. If the considered value of measured metrics has changed during the attack interval and non-attack interval, it can be concluded that the proxy server is vulnerable to the applied attack.

3.4.2 Evaluation metrics for SIP attack's effects

In this subsection we define some metrics for evaluating vulnerability of the entity against applied attacks. These criteria can help to recognize the existence of the vulnerability.

1. Completion Call Rate

In the attack period due to the heavy traffic to the proxy server, vulnerable server does not have enough resources to create new calls with legitimate users. As a result Completion Call Rate during attack period is decreased than non-attack period. Thus reducing Completion Call Rate during attack period is one of the criteria that determine the vulnerability of the proxy server.

2. Retransmission Call Rate

In attack period due to applying many requests from attacker to the victim, there is not enough time to respond to requests. When the query timeout, retransmission will be performed. So if the server is vulnerable, retransmission call rate will be considerable. The Ratio of retransmission call rate in attack period to non-attack period determines the vulnerability of the proxy server.

3. Response Time

Response time is time interval between sending a request and receiving its response. In attack period due to applying heavy traffic to victim if the proxy server is vulnerable, the response time to legitimate user request will be longer. Prolonging response time in attack period than non-attack period certify the proxy server is vulnerable against applying attack.

4. Call Set Up Time

Call set up time refers to the period of time that a request to establish a call is sent until the call is ended. In attack period due to applying heavy traffic to victim if the target is vulnerable, call set up time will be longer. Prolonging call set up time certify the target is vulnerable against applying attack.

5. Round Trip Time[1]

R.T.T is the time required for a 32-byte packet to travel from a specific source to a specific destination and back again. As stated before in attack period due to heavy traffic the target being busy, so R.T.T become longer if target is vulnerable.

3.5 Report Module

This module is responsible for comparing measured criteria in attack period with non-attack period. If these two values have significant difference, it can be concluded that the SIP entity is vulnerable to applied attack.

4. Experiment Setup and Evaluation

Because of importance and mandatory role of proxy servers in SIP environments, this entity is selected in our

[1] R.T.T

experiments. SIP proxy is more vulnerable to security threats especially against Denial of Service attacks.

Our experiment test-bed (as shown in Figure 6) consists of a user agent client (UAC), a SIP proxy server and a user agent server (UAS). The UAC and UAS run SIPp [16] for generating SIP normal traffic with selected parameters. UAC and UAS are connected by proxy server. In fact the connection between UAC and UAS as normal traffic is always available during the simulation period. Since we measure our performance metrics in field just before attack period, we did not have any consideration for normal traffic because it is not important in our experiments. Proposed scanner that is shown as pen-tester in a separate computer in figure 6 applies selected attacks to the proxy server. The pen-tester uses VoIP hacking tools of Linux Backtrack 5 on his computer which run it in a virtual machine in our experiments. Depending on the attack scenario, the proposed scanner sometimes needs a partner to design an attack against the proxy server. This partner is shown as co-pen-tester in figure 6. The generated attacks are applied to the victim proxy server from pen-tester station during the test period, passing traffic through the victim proxy server is captured and defined metrics are measured for captured traffics.

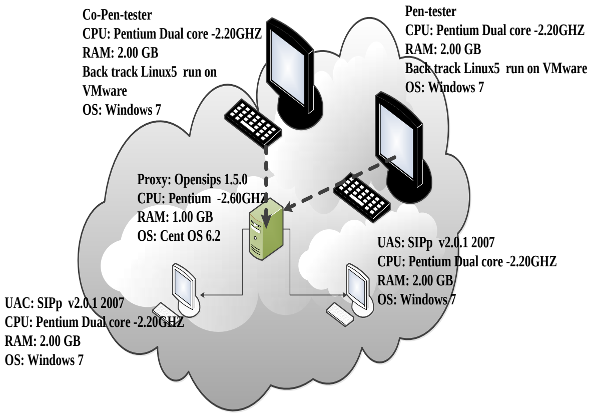

Fig. 6 Test-bed for evaluating proposed scanner

4.1 Generated attacks in operation module

As stated before, operation module is responsible for applying bandwidth depletion, memory depletion and CPU depletion attacks. As proxy server is selected among SIP entities for our test, so applied attacks should have effects on proxy servers. Therefore applied attacks are:

1. Bandwidth Depletion Attacks

a. Invite flood attacks

In this type of attack we generate large number of INVITE packets by SIPp tool. In this attack scenario we want to just deplete the bandwidth of the proxy server.

b. UDP flood attacks

UDP flood attack will produce by Hping tool. A large number of UDP packets are sent to the proxy server, so its bandwidth will be occupied with spurious packets.

2. Memory Depletion Attacks

a. Brute force attacks

SIP has a session control mechanism in the application layer. The SIP sessions consist of two different concepts: transaction and dialog. Almost all stateful SIP proxies are implemented in the transaction level and for this reason maintains all related statistics of sessions until its expiration. The attacker uses this mechanism to deplete the memory of the proxy server by routing packets to it in a rate which is more than the proxy's capacity. In other word, the pen-tester sends messages for generating call to the victim proxy server but does not responds their responses from the victim, therefore victim proxy server is made to keep the call's information for a longer period of time until time to get the response runs out. So each message leads to occupy memory more than usual. In this case proxy server's memory will be occupied and there is not enough memory to meet demands of legal users. SIPp tool with appropriate scenarios is used to generate such attacks.

b. SYN flood attacks

In this type of attack pen-tester (attacker) sends many SYN packets to the victim proxy server. The proxy server thinks TCP connection will be established therefore stores calls' information but pen-tester will not answer proxy any more. In this way proxy has to keep calls' information up to its predefined time out of the RFC 3261 (up to 180 seconds). As a result proxy's memory will be occupied by sending a large amount of SYN packets to it and will not have sufficient space for legitimate users. SYN flood attack is generated by Hping tool in our framework.

c. Incomplete transaction attacks

Pen-tester for applying incomplete transaction attacks needs coworker (co-pen-tester). In order to produce such an attack; pen-tester sends a message (e.g. an INVITE message) to the proxy server and asks him to forward it to co-pen-tester. But co-pen-tester is configured to not respond the proxy server. In consequence proxy server has to keep transaction's information until time runs out. Then proxy sends time-out to pen-tester. There are two possible cases:

Case1: When pen-tester receives time-out from the proxy, sends ACK message to the proxy. By producing a large number of this message (INVITE message), proxy's memory will be occupied and there is not sufficient space to perform other users' requests. This case is shown in Figure 7.

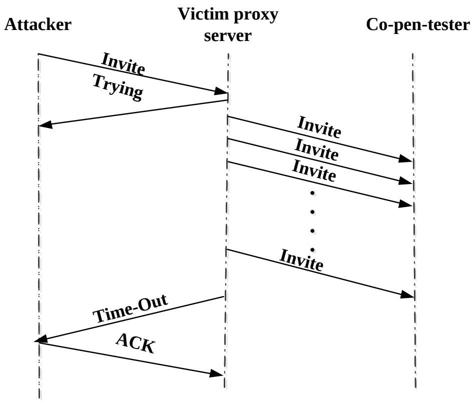

Fig.7 First case of incomplete transaction attacks

Case2: When pen-tester receives time-out from the proxy, does not respond to proxy. So proxy must repeat time-out message until specific time. In this case proxy keeps transaction information in its memory longer than usual. This case is shown in figure 8.

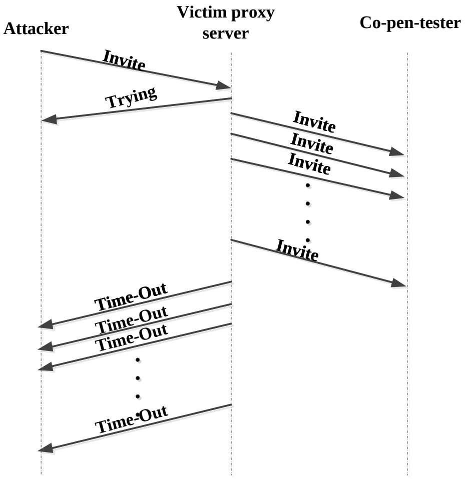

Fig.8 Second case of Incomplete transaction attacks

Both of these attacks are generated by appropriate use of SIPp in UAS and UAC modes.

d. Incomplete transaction with partner

For implementing this attack Pen-tester sends a message for example INVITE message to proxy and wants him to forward it to co-pen-tester. Co-pen-tester is configured to send TRYING messages in order to respond proxy server and made proxy to keep waiting for giving response from it. When time out to respond, victim proxy server sends time-out to pen-tester. But pen-tester does not respond by ACK to proxy. Therefore proxy repeats time-out to pen-tester for a limited time. In this case poor proxy has to keep transaction's information longer than before. This attack is shown in Figure 9.

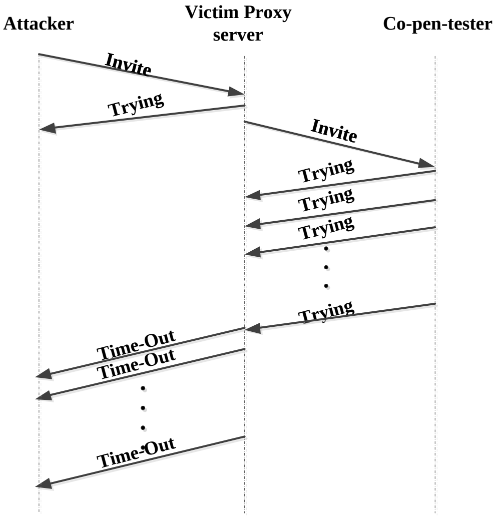

Fig. 9 Incomplete transaction with partner

3. CPU Depletion Attacks

a. ICMP flood attacks

ICMP flood attack sends a large number of ICMP packets to the victim proxy server. A lot of processing power is needed to analyze these packets. Therefore CPU will not have enough time to process requests from legitimate users. A special java application is written for implementation of this kind of attack.

b. Authentication misuse attacks

Most of the SIP servers are configured to authenticate users before their registration. The mandatory authentication mechanism of SIP is HTTP digest method which is based on the challenge and response. The attacker tries to deplete the processing power of SIP proxy by misusing the authentication process. For generating this kind of attack, the pen-tester sends a large number of INVITE messages on victim proxy server. Proxy server for each message designates a random number called nonce (for generating nonce CPU will be involved) and send back both the message and its nonce to pen-tester. The proxy expects from pen-tester for sending message and its nonce to him again. But smarty pen-tester will not do anything! By generating a large number of these messages proxy server' CPU will be busy just to generate random numbers (challenge) and will not have sufficient time to process legitimate users' requests. More details about these attacks can be found in [10].

It should be noted that all the test scenarios are about 30 seconds. During this period, the normal traffic is available on the proxy server and in the time interval [10, 12], the attack scenario is applied to the proxy server. The proposed scanner according to traffic underlying through the victim proxy server and measuring defined criteria, diagnoses the vulnerability of the proxy server in a black box manner.

4.2 Measuring evaluation criteria for analyzing vulnerability of SIP entity

In section 2-4-3 evaluation criteria to diagnose vulnerable SIP entity is expressed. Now this section mentions how to calculate them.

1. Completion call rate in attack and non-attack period

The number of completed calls in a considered window can show the activity ratio of server. For calculating the number of completed calls, those messages should be considered that at least one of the features of source tag, destination tag and call-ID have changed. The reason is that unique triple of [source tag, destination tag, call-id] specifies a dialog. The number of dialogs shows the number of completed calls. Then the average of completed calls in attack and non-attack period are measured. By comparing these two numbers in other words by reducing this number in attack period than non-attack period can be diagnosed vulnerable proxy server against applied attack.

2. Retransmission call rate in attack period and non- attack period

Messages that have same transaction identifiers [source tag, from tag, call-Id, via, CSeq[1]] have been retransmitted. To calculate number of retransmitted calls in traffic through proxy server, those messages that have same five features are counted in both attack period and non-attack periods. Then the average of retransmission calls in both periods are calculated. Increasing this number in attack period than non-attack period indicates the vulnerability of proxy server against applied attack.

3. Response time in attack and non-attack period

SIP INVITE message response time is time interval between SIP INVITE and its RINGING message. For each INVITE message response time is calculated in both attack period and non-attack period. Then the average of them in both periods are measured. By comparing these two numbers the vulnerability of proxy server against applied attack can be identified. If proxy server is vulnerable, response time will be increased in attack period than non-attack period.

4. Call set-up time in attack and non-attack period

Call setup time for INVITE message is time interval between INVITE and its ACK message. These two messages must have same call-ID. In this way we calculate setup time in attack period and non-attack period then we measure the average of call setup time in both attack and non-attack period. By comparing these two numbers vulnerability of proxy server can be identified. Increasing call setup time in attack period than non-attack period indicates that the proxy server is vulnerable to applied attack.

5. R.T.T in attack period and non-attack period

R.T.T is another criterion that is defined to detect vulnerable proxy server. For calculating R.T.T simply can use “ping” command, R.T.T is one of information in ping command. We calculate R.T.T in both attack and non-attack period. Then the average of these time intervals in both periods are calculated. If proxy server is vulnerable, the average of R.T.T in attack period will be increased.

4.3 Simulation results and analysis

We assumed that we want to check the vulnerabilities of the SIP proxy server (OPENSIPS 1.5.0). Therefore proposed scanner is applied to OPENSIPS [17] and the defined criteria are measured. Table 1 shows the results of applying bandwidth depletion attacks on OPENSIPS.

Table 1. The Results of applying bandwidth depletion attacks on OPENSIPS

OPENSIPS				
Metric	Invite Flood		UDP Flood	
	Attack	Non-attack	Attack	Non-attack
Completed Call (CC)	309	362	314	254
Retransmission Calls (RC)	8898	0	400	0
Response time (ms) (RT)	189.13	2.01	33.37	2.91
Call set up time (ms) (CST)	57.37	3.23	39.36	18.36
Round Trip Time (ms) (RTT)	1.3	0.73	9.40	0.74

Shown in Table 1, since there is a significant difference between metrics like response time and call setup time during attack period and normal period, we can conclude that the studied proxy server is vulnerable to bandwidth depletion attacks. It should be said that we assure about the vulnerability of OPENSIPS to bandwidth depletion attack by sending high volume traffic to this proxy server before starting this experiment.

Table 2 shows the results of applying memory depletion attacks on OPENSIPS.

Table 2. The Results of applying memory depletion attacks on OPENSIPS

OPENSIPS								
Metric	SYN Flood		Brute Force Attack		Incomplete Transaction		Incomplete Transaction with partner	
	Attack	Non-Attack	Attack	Non-Attack	Attack	Non-Attack	Attack	Non-Attack
CC	288	297	143	297	685	724	524	767
RC	24	0	5386	0	14766	173	47233	173
RT	58.34	2.25	86.50	4.71	6.23	5.14	40.87	1.86
CST	57.81	3.23	91.46	2.95	429.81	3.25	890.20	4.33
RTT	24.12	0.67	14.77	0.87	1.52	0.84	35.00	10.44

According to Table 2 OPENSIPS is vulnerable to memory depletion attacks. For example call set up time measured in SYN flood attack is increased in attack period than non-attack period. As a result, OPENSIPS is vulnerable against SYN flood attack.

[1] Call sequence

Table 3. The Results of applying CPU depletion attacks on OPENSIPS

OPENSIPS				
Metric	ICMP Flood		Security Checking Attack	
	Attack	Non-attack	Attack	Non-attack
CC	288	297	2204	12196
RC	24	0	6220	0
RT	58.34	2.25	107.37	2.80
CST	57.81	3.23	736.78	5.62
RTT	24.12	0.67	3.14	0.65

According to Table 3 OPENSIPS is vulnerable to CPU depletion attacks because of significant differences between defined metrics in attack period and non-attack periods.

4.4 Validating proposed framework

The purpose of this section is to demonstrate the validity of the proposed framework. As stated before proposed framework's duty is extraction of entity's vulnerabilities. So in other words we want to show that our proposed framework correctly identifies the exist vulnerabilities in an entity. Figure 10 shows steps to prove framework functionality correctness.

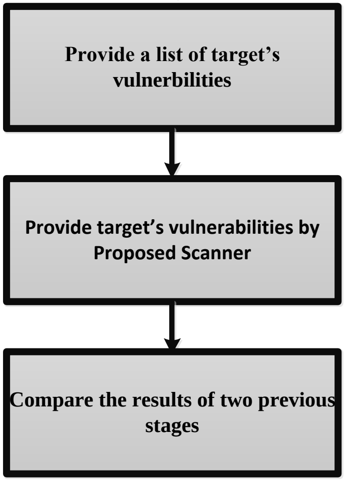
Provide a list of target's vulnerbilities
Provide target's vulnerabilities by Proposed Scanner
Compare the results of two previous stages

Fig. 10 steps to prove framework functionality correctness

1. Providing a list of target's vulnerabilities

First we should find a list of vulnerabilities of the target. In other words we just answer the question that the proposed system seeks to answer. For providing the list of vulnerabilities we do as follows:

We consider some resources in victim that our attacks can have effects on it. For example for memory depletion attacks, we consider memory. Then we apply the attack that has effects on those resources in a very high rate. If the victim becomes unavailable we can conclude that the victim is vulnerable to that attack. We do this experiment on OPENSIPS 1.5.0 and found out that OPENSIPS is vulnerable to all of our applied attacks. This step is shown in Figure 11.

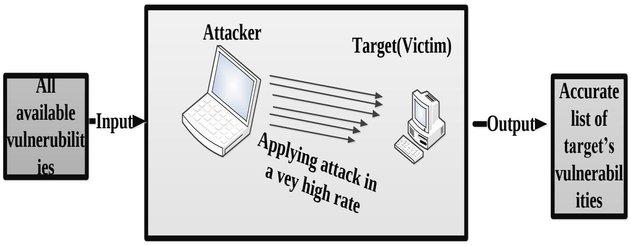
All available vulnerubilit ies
Input
Attacker
Target(Victim)
Applying attack in a vey high rate
Output
Accurate list of target's vulnerabil ities

Fig. 11 providing a list of target's vulnerabilities

2. Providing target's vulnerabilities by the proposed scanner

In this step we apply our scanner to our victim to get the list of vulnerability the victim has. We did this experiment in section 4-3 and concluded that our victim (OPENSIPS 1.5.0) is vulnerable to all of our applied attacks. This step is shown in Figure 12.

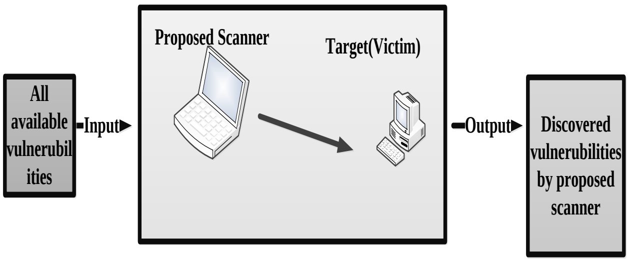
All available vulnerubil ities
Input
Proposed Scanner
Target(Victim)
Output
Discovered vulnerubilities by proposed scanner

Fig. 12 providing target's vulnerabilities by the proposed Scanner

3. Compare the results of two previous steps

In this step the results of two previous steps are compared. If these results are same, we can conclude that proposed scanner operates correctly. These two results are same so our scanner detects vulnerabilities correctly.

5. Conclusions

Although the main focus of this paper is on VoIP and especially on SIP protocol, the vulnerability analysis framework presented in this paper is general to accommodate a variety of similar protocols. In this paper, a SIP security assessment framework is presented by using the idea of penetration testing. In this context, producing non-destructive attacks is used to identify vulnerabilities in SIP entities. The main idea of this paper includes the identification of vulnerabilities, defining criteria for assessing vulnerability and the generating of non-destructive and controlled attack. To evaluate the proposed idea, a practical VoIP test bed is used.

Acknowledgment

The authors would like to thank Iran Telecommunication Research Centre (ITRC) for its financial and scientific support.

References

[1] Angelos D. Keromytis, Senior Member, “A Comprehensive Survey of Voice over IP Security Research”, IEEE Communications Surveys and Tutorials, Vol 14, No. 2, pp. 514-537, 2012

[2] Angelos D. Keromytis, “Voice over IP: Risks, Threats and Vulnerabilities”, In Proceedings (electronic) of the Cyber Infrastructure Protection (CIP) Conference. June 2009

[3] Angelos D. Keromytis, "Voice over IP Security: Research and Practice", IEEE Security & Privacy, Volume: 8 , Issue: 2 , pp 76- 78, March-April 2010

[4] Angelos D. Keromytis, “A Look at VoIP Vulnerabilities”, login: Magazine, vol. 35, no. 1, pp. 41 - 50, February 2010.

[5] “An Overview Of Vulnerability Scanners”, The Government of the Hong Kong Special Administrative Region, online document: http://www.infosec.gov.hk/english/technical/files/vulnerability.pdf, 2008

[6] Andrew Whitaker, Daniel P. Newman, “Penetration Testing and Network Defense”, Cisco Press, November 04, 2005

[7] H. Abdelnur, R. State, I. Chrisment, C. Popi, “Assessing the security of voip services”, 10th IFIP/IEEE International Symposium on Integrated Network Management, 2007, pp. 373 – 382

[8] Peter Steinbacher, Florian Fankhauser, Christian Schanes, Thomas Grechenig, “Work in Progress: Black-Box Approach for Testing Quality of Service in Case of Security Incidents on the Example of a SIP-based VoIP Service”, IPTComm 2010, 2-3 August, pp. 101-110

[9] H. Abdelnur, V. Cridlig, R. State and O. Festor, “VoIP Security Assessment: Methods and Tools”, VoIP MaSe 2006, pp. 28-32

[10] Z. Asgharian, H. Asgharian, A. Akbari, B. Raahemi, *"A framework for SIP intrusion detection and response systems"*, International Symposium on Computer Networks and Distributed Systems (CNDS), pp.100-105, 2011

[11] S. Ehlert, D. Geneiatakis, T. Magedanz, *“Survey of network security systems to counter SIP-based denial-of-service attacks”*, Computers & Security, 29(2), pp. 225-243, 2010

[12] Free Security Scanner for Network Exploration and Security Audits, http://nmap.org/

[13] small comand line tool for developers and administrators of SIP applications, http:// www.voip-info.org /wiki /view/Sipsak.

[14] Active Network Security Tool, http://www.hping.org/

[15] National Vulnerability Database, http://nvd.nist.gov/

[16] free Open Source test tool / traffic generator for the SIP protocol, http://sipp.sourceforge.net/

[17] Open Source SIP proxy/server for voice, video, IM, presence and any other SIP extensions. http://www.opensips.org/

[18] Dorgham Sisalem, John Floroiu, Jiri Kuthan, Ulrich Abend, Henning Schulzrinne, “SIP SECURITY”, John Wiley & Sons Ltd, 2009

[19] R. Islam, S. Ghosh, *"SIP Security Mechanism Techniques on Voice over Internet Protocol (VoIP) System"*, International Journal of Computer Applications in Engineering Sciences (IJCAES), Vol I, Issue I, March 2011

[20] A. D. Keromytis, “Voice over IP Security, A Comprehensive Survey of Vulnerabilities and Academic Research”, Springer Briefs in Computer Science, 2011

[21] S. McGann, D. C. Sicker, *"An Analysis of Security Threats and Tools in SIP-Based VoIP Systems"*, online document: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.94.9378, 2006, Last Access: 21-Sept-2012

[22] F. Fankhauser, M. Ronniger, C. Schanes, T. Grechenig, *"Security Test Environment for VoIP Research"*, International Journal for Information Security Research (IJISR), Volume 1, Issue 1, March 2011

Mitra Alidoosti received both her B.Sc. and M.Sc. in Computer Engineering from Computer Engineering Department of Iran University of Science and Technology (IUST) in 2009 and 2012. Currently she is a PH.D candidate in computer engineering. Her research interests are computer network security, penetration testing in computer networks and web pentesting.

Hassan Asgharian received his M.Sc. in Computer Engineering from Amirkabir University of Technology (AUT) in 2009 and He also has got his B.Sc. from Computer Engineering Department of Iran University of Science and Technology (IUST) in 2006. Currently he is a PH.D candidate in computer engineering in IUST and working as a research assistant in the Network Research Group (NRG) of the Research Center of Information Technology (RCIT) of Computer Engineering Department in Iran University of Science and Technology.

Ahmad Akbari received his Ph.D. in Electrical Engineering from the university of Rennes 1, Rennes, France, in 1995. Dr. Akbari is currently an associate professor at Iran University of Science and Technology, Iran. Dr. Akbari works on Speech Processing related research topics (especially speech enhancement) for more than 20 years. His current research interests include Intrusion Detection and Response Systems, VoIP Communications, Next Generation Networks, VoIP and SIP Security and also Data Communications Networks. Dr. Akbari’s work has appeared in several peer-reviewed journals and conference proceedings.