Volume 3, Issue 1, January 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

The Protection Mechanism against DOS and SQL Injection Attack in SIP Based Infrastructure

Harish C. Sharma
Dept. of Computer Science
SGRR-ITS, Dehradun
India

Sanjay Sharma
Dept. of Computer Science
SGRR-ITS, Dehradun
India

Sandeep Chopra
Dept. of Computer Science
SGRR-ITS, Dehradun
India

Pradeep Semwal
Dept. of Computer Science
SGRR-ITS, Dehradun
India

Abstract: Session Initiation Protocol (SIP) is a protocol for signalling multimedia sessions with one or more participants. SIP is used to establish session. SIP is similar to HTTP protocol which is also a text based protocol. The paper presents some of the common security vulnerabilities of the Session Initiation Protocol and presents possible solution for protecting against different threat such as denial-of-service and sql injection.

Keywords: Sip security, DoS, SQL injection, Vulnerabilities, sip nonce

I. Introduction

Voice over Inter Protocol (also called VoIP, IP Telephony, Internet Telephony, and Broadband Phone) is the routing of voice conversations over the Internet or through any other IP-based network. VoIP allows one to use a single high speed internet connection for all voice, video, and data communications. The process of VoIP is dependent on signaling and media transport. A signaling protocol, such as SIP [1] performs the legwork: locating users call parameters, modifications and building or ending a session. SIP was developed by Internet Engineering Task Force (IETF) in 1999 and later defined by J.Roseberg et al, by rfc 3261 in june 2002.

A session can be voice, video or instant messaging and is described by SDP. SIP is a text based protocol, using UTF-8 encoding, very much like HTTP or SMTP that can establish in IP network. SIP thus enables service providers to integrate basic IP Telephony services with web, Email, and chat services. In short, SIP promises to enrich consumer's live through a host of innovative services, such as audio, videoconferencing, interactive gaming, and voice-enrich e-commerce. The paper describes different security mechanism available with the SIP protocol such as HTTP digest, S/MIME, TLS and IPSec.

II. SIP Components

- User agent client
- User agent server
- Proxy Sever
- Redirect Server
- Registrar Server

User agent client:SIP User Agent is a logical entity at VoIP endpoints capable of initiating and answering calls between peers. SIP user agent consists of User Agent Client UAC is one of two client–side components, the other being the User agent server (UAS). The UAC is an application that initiates up to six feasible SIP requests to a UAS. The six requests issued by the UAC are: INVITE, ACK, OPTIONS, BYE, CANCEL and REGISTER.

User agent server:UAS is the Server that hosts the application responsible for receiving the SIP requests from a UAC, and on reception returns a response to the request back to the UAC. The UAS may issue multiple responses to the UAC, not necessarily a single response. Communication between UAC and UAS is client/server and (peer-to–peer).

The Proxy server: The proxy server acts as an mediater it forwards services, requests to other UASs or UACs for servicing. Proxy server can use an intraorganizational configuration through which to route all its sip communications. Intra-organizational configuration can be described when users messages are routed through a proxy server before the messages are relayed to the destination SIP client.

Redirect Server: Accept SIP request map into new addresses and return these addresses to the client. They do not initiate request and do not accept calls. Hence the less overhead are required as compared to the proxy server.

Registrar Server: The Registrar server makes it possible for users to alter the address at which they are contactable. This is possible through the SIP client sending a REGISTER request of change of an address to the registrar server, which then accepts the request and records the user's new address. There are two ways in which the SIP clients can contact the registrar server. The first way is through a direct approach, by utilizing information that is configured into the client. Secondly through an indirect approach, which users the multicast address to contact the registrar server.

III. SIP Architecture

The SIP[3] protocol itself is modeled on the three-way handshake method implemented in TCP. We will consider the setup here when a proxy server is used to mediate between endpoints. The process is similar with a redirect server, but with the extra step of returning the resolved address to the source endpoint. During the setup process, communication details are negotiated between the endpoints using Session Description Protocol (SDP), which contains fields for the codec used, caller's name, etc. If "X" wishes to place a call to "Y" he/she sends an INVITE request to the proxy server containing SDP info for the session, which is then is then forwarded to Y's client by X's proxy, possibly via her proxy server. Eventually, assuming Y wants to talk to X, he/she will send "OK" message back containing her call preferences in SDP format. Then X will respond with an "ACK". SIP provides for the ACK to SDP instead of the INVITE, so that an INVITE may be seen without protocol specific information. After the "ACK" is received, the conversation may commence along the RTP[3] ports previously agreed upon.[4][5][10].

Method	Purpose
INVITE	Initiatate a session
ACK	Acknowledge session initiation
OPTIONS	Query server capability
BYE	Terminate
CANCEL	Cancel pending request
REGISTER	Register user location

Table 1: SIP request messages

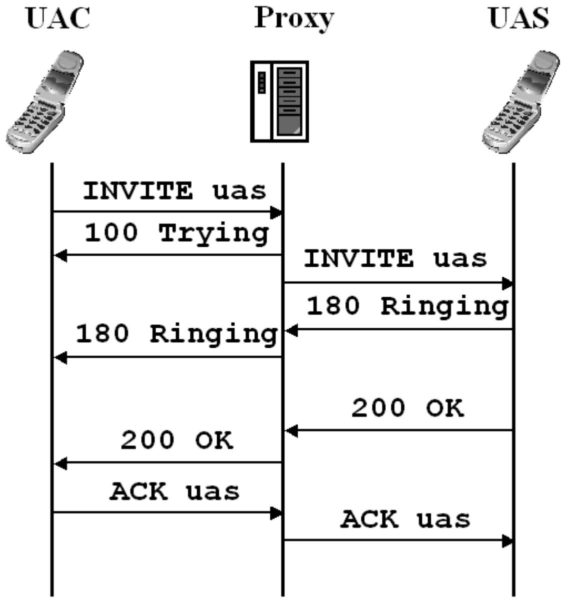

Figure 1 : SIP call scenario

IV. The Problem Definition

SIP runs over UDP, provides opportunities for attacks and vulnerabilities like Denial of service, SQL injection, Tearing down session and hijacking. Attackers on the network may also be able to modify packets, perhaps at some compromised intermediary node.
[3] [6]

1. Denial of Service

The Denial of Service attack are explicit attempt to disable the target machine and preventing user to use its services. The DoS can take form malformed packets, manipulating SIP states and simple flooding, such as a "REGISTER" or "INVITE" storm (a flood of packets. DoS can also be directed at a firewall. SIP requires management of UDP ports for media. A DoS attack that floods the firewall with calls can prevent it from properly managing ports for legitimate calls[2]. Attackers can create bogus requests that contain a falsified source IP address and a corresponding Via header field that identify a targeted host as the originator of the request and then send this request to a large number of SIP network elements, thereby using hapless SIP UAs or proxies to generate denial-of-service traffic aimed at the target [1].

***Malformed signaling*:** Unusually long or syntactically incorrect SIP message packets, referred to as "malformed", are sent to the UA degrading its performance, resulting in its inability to process normal setup and teardown messages for calls.

***Invalid call setup messages*:** A number of invalid call set up messages, such as a SIP **ACK** request when none is expected, are sent to cause the endpoint to crash, reboot, or exhaust all of its resources[8].

The DoS attack is targeted to consume resource such memory and cpu and bandwidth.

2. SQL Injection Attack

In order to store and administer user credentials and appropriate data for providing value-added services to end-users, SIP relies on databases such as MySQL, Postgress , or Oracle . This fact makes SIP-based services and specifically any authentication procedure vulnerable to attacks similar to a known Internet attack known as SQL Injection. SQL injection in SIP can be triggered every time a SIP network entity (e.g., SIP UA, SIP Proxy) asks for authentication. So, considering the case where a SIP network element requests authentication, the UA on behalf of the authorized user computes the appropriate credentials based on the HTTP Digest mechanism. The result of this computation age's authorization header. Then the message is forwarded to the SIP proxy server, which must authenticate the received message. It recalculates the user's credentials using the user's password stored in the "Subscriber" table. To accomplish this task, it generates an SQL statement of the following syntax: [4]

SELECT password FROM subscriber WHERE username=' sgrr' AND realm='192.168.10.23'

In the case where a malicious user tries to launch an attack in the SIP architecture by exploiting SQL injection, he/she spoofs the SIP message and inserts the malicious SQL code in its Authorization header. This message can be any
SIP message requiring authentication by a SIP server. The code can be embodied either in the username or in realm fields in the Authorization header.

As soon as the proxy receives a SIP message with an infected Authorization header, it generates and executes the following SQL statement:

SELECT password FROM subscriber WHERE username='sgrr;

UPDATE subscribe SET first_name='malicius'
WHERE username='sgrr'

As a result, message authentication fails, but the second command manages to change 'sgrr's first_name' to 'malicious'. It is also possible for a malicious user to attempt to employ similar SQL commands, aiming to make the database service useless and cause a DoS to the provided
VoIP service.

V. Defense Mechanism

(i) The Denial of Service (DoS) attack are one of the major threat users facing on the Internet. There is a unique field nonce in the SIP, 407 (Authentication Required) message which can be used to avoid replay attacks. A nonce is a server-specified data string which should be uniquely generated each time a 401(Unauthorized) or 407 response is made. Since most servers have a timer to mitigate replay attacks, the servers do not have to keep a record of the nonce. Thus, it is important for the firewall to perform the nonce checking. When an INVITE/REGISTER messages is received at port, the firewall will check the SIP header, looking for a "nonce" value. If the incoming SIP request does not have a nonce value, the firewall generates a nonce value, which should be the result of a cryptographic secret function computed over the CallID and source IP address. This ensures that the nonce is unique for each session, as a new CallID is generated when a session is initiated[7].
The firewall will then send back a 407/401(AuthenticationRequired/Unauthorized) message back to the client, with the calculated nonce value and drop the session. After the client receive a 407/401 message, it resends an INVITE message with the same CallID, serverspecified nonce, username and password.

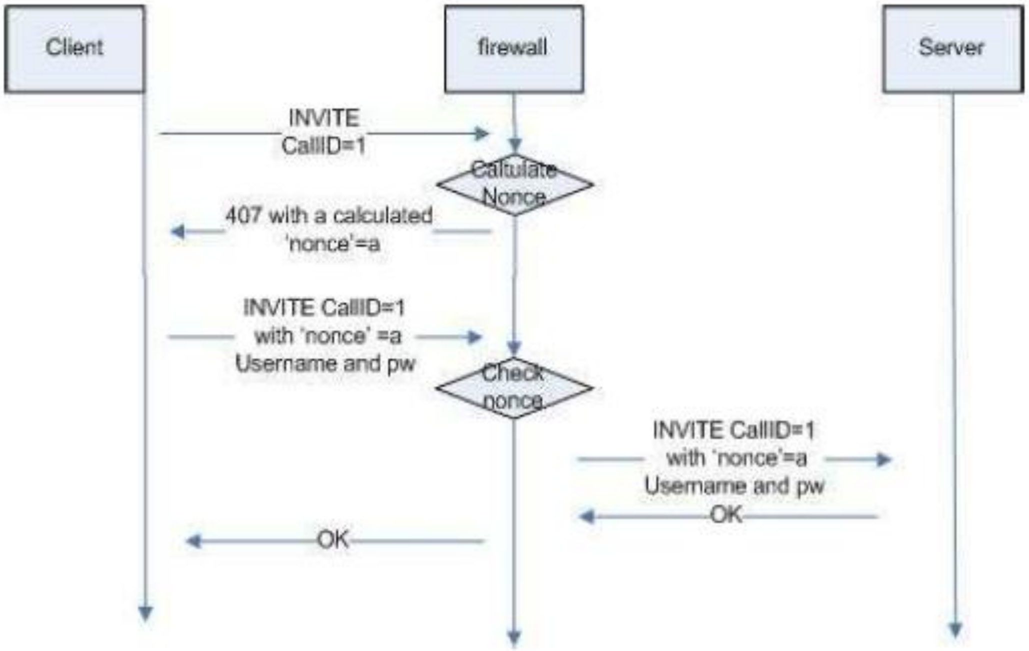
Client
firewall
Server
INVITE CallID=1
Calulate Nonce
407 with a calculated 'nonce'=a
INVITE CallID=1 with 'nonce' =a Username and pw
Check nonce
INVITE CallID=1 with 'nonce'=a Username and pw
OK
OK

Figure 2 : Firewall checking nonce

When the firewall receives an INVITE with a nonce value, it will extract the nonce value, recalculate and compare it with the CallID and source IP address. If it matches, the request is passed to the server. For better attack mitigation, the cryptographic secret should be changed after a small period of time, e.g. 30 seconds. This provides a stateless authentication as the firewall does not need to store multiple CallIDs and nonce entries in a database, and it can protect the server from DoS, Spoofed SIP flooding attacks.

(ii) The mutual authentication of proxy servers through mechanisms such as TLS significantly reduces the potential for rogue intermediaries to introduce falsified requests or responses that can deny service. This commensurately makes it harder for attackers to make innocent SIP nodes into agents of amplification [1][9][11]

Nonce Setting

In order to prevent the DoS attack from SIP server the small nonce can be generated by the SIP server which expires in a very short period of time. Or it does wait for a long peroid of time for user to input values. However, it can frustrate the end users. For instance, if a poor network connection renders a large packet round trip time, the nonce issued to authorized user can expire before the second request packet arrive at server. The same problem applies to the threshold of the number of SIP requests from each IP address. A relatively low threshold can throttle a small scale DoS attack. However, it could also block the SIP requests from a proxy that aggregates the SIP requests from multiple SIP agents. Consequently, these parameters need to be set with great care to optimize the server's performance.

Iancu Algorithm

(i) Iancu (2003) developed a DoS flooding mitigation mechanism dubbed ‘‘Pike’’ that rate-limits incoming traffic on a per-host basis. This method is listed as an example for the various ratelimiting software mechanisms available as add on for SIP servers or in commercial security solutions. The algorithm counts all incoming requests per IP address in a defined time frame. Whenever a fixed upper limit is reached, further messages from the offending IP address are not processed for a limited time. [12]

(ii) The technique for preventing SQL tampering in SIP is to digitally sign the messages that are exchanged. As a result, any modification in a SIP message can be detected, having the message automatically discarded by the SIP server. Generally, digital signatures can protect SIP messages from any sort of tampering attack. Nevertheless, digital signatures scheme requires the installation of a global or layered Public Key Infrastructure (PKI) beforehand. Moreover, this method is totally

ineffective against “insiders”. Finally, in order to avoid errors in input validation or to prevent any other malicious attempt, the SQL account that the SIP server uses to connect to the database must have only the minimum-required privileges.
(iii) The other solution to secure an application against SQL injection, developers must never allow client-supplied data to modify the syntax of SQL statements. In fact, the best protection is to isolate the web application from SQL altogether. All SQL statements required by the application should be in stored procedures and kept on the database server. The application should execute the stored procedures using a safe interface such as JDBC’s CallableStatement or ADO’s Command Object. if arbitrary statements to be used, use PreparedStatements. Both PreparedStatements and stored procedures compile the SQL statement before the user input is added, making it impossible for user input to modify the actual SQL statement. [13]

Some other preventive security measures for SIP

- Support TLS and other standards-based security where possible.
- Perform SIP-aware NAT and media port management.
- Perform granular Call Admission Control (CAC). Control the number of simultaneous calls.
- Monitor for unusual calling patterns.
- Provide detailed logging of all SIP messages. Log everything for non-authenticated calls.
- Monitor for external registration hijacking attempts.

VI. Conclusion

Security is divided into the aspects of authentication, confidentiality, integrity and availability. SIP has threats in each of these aspects. SIP system can be best secured by best practices for securing VoIP and using standards-based security on all system components. These same security standards should be used as SIP is exchanged with components in an untrusted network. Use SIP-optimized firewalls, which both support use of standards-based security and provide the best possible protection where system-wide standards-based security is not possible.

References

[1] Rosenberg, et. al. “*Session Initiation Protocol*”, RFC 3261, June 2002

[2] Gaston Ormazabal, Sarvesh Nagpal, Eilon Yardeni, and Henning Schulzrinne, *Secure SIP: A Scalable Prevention Mechanism DoS Attacks on SIP based VoIP Systems*, Springer-Verlag Berlin Heidelberg 2008

[3] J. Peterson, Neustar, *A Privacy Mechanism for Session Initiation Protocol*, November 2002

[4] Dimitris Geneiatakis et al, *Survey of Security Vulnerability in Session Initiation Protocol*, IEEE Communications Surveys & Tutorials 3rd Quarter 2006

[5] Johnston, et al. “*Session Initiation Protocol Basic call flow example*”, RFC 3665, Dec 2003

[6] Mark Collier, *Basic Vulnerabilities issue, SIP Security*, March 2005

[7] Felipe Huici, Saverio Niccolini, Nico d Heureuse, *Protecting SIP against very large flooding DoS Attacks*,IEEE, Globecom2009 Proceedings.

[8] D. Sisalem, J. Kuthan, and G. Schäfer, “DoS Attacks on SIP Infrastructures,” *Voice over IP: Challenges and Solutions GlobeCom 2004*, Dallas, TX, Dec. 2004

[9] Samer El Sawda and Pascal Urien, *SIP Security Attacks and Solutions: A state-of-the-art review*, 0-7803-9521-2/06 2006 IEEE

[10] J. Bilien, E. Eliasson, and J.-O. Vatn, “Secure VoIP: Call Establishment and Media Protection,” *2nd Wksp. Securing Voice over IP*, Washington DC, June 2005.

[11] Salsano, S., Veltri, L. and D. Papalilo. SIP security issues: the SIP authentication procedure and its processing load IEEE Network, Volume:16, Issue:6 , Nov/Dec 2002. Pages: 38-44.

[12] Sven Ehlert, Dimitris Geneiatakis, Thomas Magedanz, *Survey of network security systems to counter SIP-baseddenial-of-service attacks* computers & security 29 (2010) 225 – 243

[13] Dimitris Geneiatakis et al, *SIP Message Tampering THE SQL code INJECTION attack* , IST 2004-005892