Secure SIP from DoS based Massage Flooding Attack

Md. Ruhul Islam[1], Smarajit Ghosh[2]

[1]*Sikkim Manipal Institute of Technology, Majitar, Rangpo, East Sikkim-737136*

[2]*Thapar University, Patiala-147004, Punjab*

[1]md_ruhul@rediffmail.com
[2]smarajitghosh@rediffmail.com

Abstract- **Over IP network the SIP-based VoIP system build , so it is precious by the IP network security problem. In this paper we concentrate on the issue of denial of service (DoS) attacks which targeting the hardware and software of voice over IP servers .In this situation we mainly identify attacks based on exhaustion the memory of VoIP servers, attacks on the CPU. A major conclusion is that SIP provides a wide range of features that can be used to accumulate DoS attacks. Discovering these attacks is inherently difficult, in the case of DoS attacks on other IP components. With sufficient server design, implementation and proper hardware the effects of a large portion of attacks can be reduced. Besides the server implementation and hardware we present some optimizations that reduce the contacting DNS servers using caches, policies and extensions to the SIP messages.**

***Keywords*— VoIP, SIP, DoS , Message Flooding, SMTP.**

I. INTRODUCTION

Voice over IP (VoIP) is an umbrella term for a set of technologies that allow voice traffic to be carried over Internet Protocol (IP) networks. VoIP transfers the voice streams of audio calls into data packets as opposed to traditional, analog circuit-switched voice communications used by the public switched telephone network (PSTN).

Security threats are considered minimum in current circuit switched networks. In an open environment such as the Internet, mounting an attack on a telephony server is, however, much simpler. VoIP services are based on standardized and open technologies (i.e. SIP, H.323, MEGACO) using servers reachable through the Internet, implemented in software and provided often over general purpose computing hardware. This type of services can suffer from security threats as HTTP based services. Instead of creating thousands of voice calls, the attacker can easily send thousands of VoIP invitations in a similar manner to attacks on Web servers. These attacks are simple and easy to access in internet, also cheap. Besides launching brute force attacks by generating a large number of useless VoIP calls, attackers can use certain features of the used VoIP protocol to invite higher loads at the servers. The session initiation protocol (SIP) is ever more establishing itself as the de-facto standard for VoIP services in the Internet and next generation networks.

A. Session Initiation Protocol (SIP)

Session Initiation Protocol, SIP is an application layer protocol that has been designed by Internet Engineering Task Force (IETF).It defines initiation, modification and termination of interactive, multimedia communication sessions between users. SIP has integrated elements from other protocols that are broadly used on the Internet. It is a text-based client-server protocol with almost the same structure as the HyperText Transport Protocol (HTTP [5]) and Simple Mail Transport Protocol (SMTP [5]). The structure of the protocol easy to follow and understand. The SIP messages have the same structure as the messages in HTTP and contain a request line or a status line followed by at least six header fields. After the header field there may be an attached message body, the type and size is described by some of the header fields. As for HTTP and SMTP, SIP supports the popular Multipurpose Internet Mail Extension (MIME [5]) for describing the content in the message body. In most cases the message body consists of a Session Description Protocol (SDP [8]) message. It describes the media transfer after the signaling phase. The SDP message has a MIME subtype of *application/sdp*. Even sometime SIP message body in most cases contains a SDP message also contain other MIME subtypes, like e.g. *text/plain* or *image/gif*. The client-server structure is based on a client that issues a request for a service and a server handles the request and responds with a service. A SIP-enabled end-device, SIP User Agent (UA), has both a client and server application. All requests from a client UA contain a method in their request lines. In current version of SIP [6] there exist six different methods which shown below table1:

TABLE 1: SIX DIFFERENT METHODS IN CURRENT SIP VERSION

Method name	Description
INVITE	Invites UA server to a call and establish a new connection. Can hold media capabilities.
ACK	Used to acknowledge a response generated by a received INVITE request.
BYE	Terminates the media session between two users.
REGISTER	Used for registering information about a UA clients location.
CANCEL	Terminates a non-acknowledged invitation.
OPTION	Used to get information about supported capabilities.

The responses from the UA server have the same structure as in HTTP responses. All SIP responses have a status line which contains a status code and a formal description of the response. The status codes are ordered in six different classes and each class represents a specific type of response. Most of the header fields in the response are copies of the header fields from the request that the response corresponds to. But some header fields are specified only for response use. A response may contain a message body and the MIME subtype is in most cases the same as for the corresponding request. The UA server can include a SDP message in the message body for reconciliation of the media types that should be used during the media transfer. There exist other applications than UA clients and UA servers that are able to handle SIP messages, namely:

• Proxy servers
• Redirect servers
• Registration servers
• Gateways

Proxy servers have the task of forwarding requests from the UA client to the address specified in the SIP message. Proxy servers have also the capability to modify some parts of the header in the message, e.g. the Via header field. They can also require authentication to a request before they forward the message. Responses from the UA server are also forwarded down to the client. The responses always take the same way back as the request took, i.e. forwarded through the same proxies. If the SIP UA has been configured to always send its requests through a specific proxy, then this proxy is called the outbound proxy for the SIP UA.

A redirect server does not issue any request of its own. After receiving a request the server gathers a list of alternative locations and returns a final response.

A registration server, also called registrar, has the task of tolerant REGISTER requests from a UA. The request contains information on how to reach the UA. The information is saved so other UA can ask the registrar where the UA is located. The registrar can require authentication.

B. Denial of Service

Denial-of-Service (DoS) attacks are a class of network attacks performed to interrupt or terminate applications, servers, or even whole networks, with the aim of distracting genuine users' communication. Disruption targets are web browsing, listening to online radio, or even interrupting essential communication, e.g. power plant network control traffic. DoS attacks are commonly performed intentionally and in most cases difficult to counter. In most of cases it is only possible to moderate, but not to completely prevent the attack.

C. Vulnerabilities in Application, Protocol Stack or Operating System:

One of the common way to achieve a DoS attack is to exploit vulnerabilities in a software component on the target machine. This includes vulnerabilities in application servers, network stacks, or general operating system vulnerabilities. Vulnerabilities in massive projects are a common case, it is impossible to predict every time where software is deployed. To exploit the vulnerability, an attacker sends a messages crafted in a specific way that takes advantage of that given vulnerability.

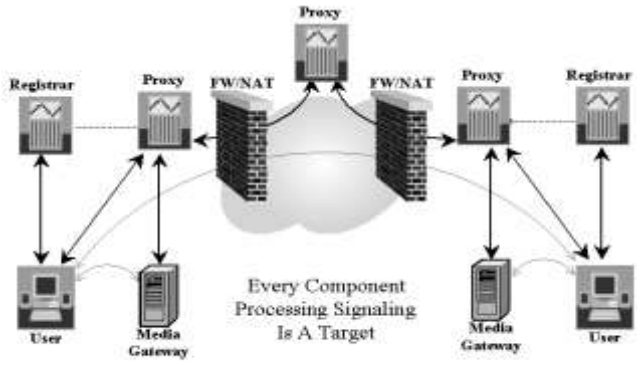

Fig 1: DoS attack process on SIP based IP network

DoS can take the form of malformed packets, manipulating SIP states, and simple flooding, such as a "REGISTER" or "INVITE" storm (a flood of packets).

DoS can be especially damaging if your key voice resources are targeted (e.g., media gateways, AA, IVR, VM, and other systems). DoS can also be used to generate large numbers of toll, information (411), or emergency calls (911). Your network can also be used as a DoS launching point, from which the generated calls are directed at another enterprise.

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.

II. GENERAL THREATS

Security and privacy requirements in a SIP communication environment are expected to be equivalent to the currently leading communication platform, the PSTN. Divergent to the PSTN, SIP networks are commonly deployed over the open Internet, which complicates the provision of security and privacy in SIP networks.

SIP messages may contain information that a user or provider wishes to keep private. For example, message headers may expose confidential information about the communicating parties. The SIP message body may also contain user information (addresses, telephone number, etc.) that must not be exposed to third parties.

A. SIP DoS by Message Flooding

The most common hint of a DoS attack is where an opponent sends a large amount of messages to a target with the goal to crush the target's processing capabilities .This type of attacks are generally hard to detect, as the utilized attack messages are usually valid messages and it is not easily noticeable from regular messages However, the whole SIP network can be severely jeopardized if another component of the network (e.g. STUN server, RTP proxy or DNS server) is being attacked.

B. Vulnerable SIP Resources

The majority of DoS attacks are based on exhausting some of a server's resources and causing the server not to operate properly due to lack of resources. With SIP servers, there are three resources needed for operation: memory, CPU and bandwidth.

1) Memory

A SIP server needs to copy each incoming request into its internal buffers to be able to process the message. The amount of buffered data and the time period the server is supposed to keep. The buffered data varies depending on whether the server is working in a stateful or stateless mode. In any case, the server will at least need to maintain the buffered data while contacting another entity such as an AAA, DNS server or a database.

Stateless servers: Stateless servers need only to maintain a copy of the received messages while processing those messages. As soon as the destination to which a message is to be sent to is determined and the message is sent out, the server can delete the buffered data.

Stateful servers: In general we can distinguish between two types of state in SIP:

i. Transaction state: This is the state that a server maintains between the start of a transaction, i.e., receiving a request and the end of the transaction, i.e. receiving a final reply for the request. A transaction stateful server needs to keep a copy of the received request as well as the forwarded request. Typically, transaction context consumes about 3 Kilobytes (depending on message size, forking and memory management overhead) lasting about one to tens of seconds if users interaction is involved.

ii. Session state: In some cases servers may need to maintain some information about the session throughout the lifetime of the session. This is especially the case for communication involving firewall or NAT traversal, for accounting purposes or security reasons as is the case for the 3GPP architecture [8].

2) CPU

After getting a SIP message, the SIP server needs to parse the message, do some processing (e.g. authentication), carry out transaction mapping and forward the message. Depending on the type and content of the message and server policies the actual amount of CPU resources might vary, while the CPU capacity of a well engineered and configured proxy should be able to process SIP messages up to link capacity, there are many server operations which can cause blocking. Such operations may be misused to quickly paralyze the server's operation. ISIP server can take action which consumes time, memory and processing power. We can distinguish the following actions:

i. Message parsing: The content of SIP messages is coded like plain text. The SIP standard allows to give freedom in setting the order of the message headers, using small or capital letters, including line breaks and so on. Thereby the time taken to parse a message depends very much on the efficiency of the message parser as well as the content of the message.

ii. Security check: SIP might use secure protocols providing state-of-threat security (TLS, S/MIME, IPSec).The most broadly deployed security protocol is still digest authentication. One of the reasons is the cost factor: Vendors attempt to keep prices of end-devices low and build devices with limited memory capacity (typically less than 1 Megabyte) leaving themselves few space for cryptographical protocols. Aspects of DoS attacks on SSL and IPsec are independent of SIP itself.

iii. Supporting services: In the context if this document a supporting service is a service that is used by a SIP server in order to fulfill its task of forwarding a received SIP message to the correct destination. Such services include but are not restricted to:

iv. AAA servers: These are required to check the identity of a user, its eligibility for using a certain service and collecting information about used services.

v. DNS servers: A SIP server needs to contact DNS servers to resolve the SIP addresses included in the SIP messages.

vi. Application server: To execute user specified routing rules a SIP server might need to contact a CPL server or some other form of an application server. This relations is realized in general over some inter-process communication and involves in general the generation of appropriate requests, exchanging those requests over the network and analyzing the reply.

3) Bandwidth

When the maximum access links would try to connect a SIP server to the Internet, at those level congestion losses may happen. By overloading the servers access links one could cause the loss of SIP messages which causes session setup times might be taken long or even the failure of session setups.

Bandwidth deletion attacks can be targeted at any host that has a TCP/IP stack regardless of the applications running on the host. Hence, while these attacks do not target any SIP-specific characteristics they can be launched at any SIP component. Protection of bandwidth is a general transport-layer issue unspecific to SIP.

C. DoS Attacks Based on Exhaustion of Memory

SIP servers is one of the easier targets for DoS attacks. Measurements indicate that a stateful server flooded with a continuous stream of requests belonging to different transactions will run out of memory very quickly.

Brute force attacks: The simplest method for mounting an attack on the memory of a SIP server is to start a large number of SIP sessions with different session identities. Different SIP messages can be used for message flooding, e.g. REGISTER, INVITE, or OPTIONS.

1) Incomplete transactions:

With brute force attacks, a system's memory is exhausted by creating a larger number of transactions. With the incomplete transactions attack, the attacker prolongs the lifetime of transactions. It is similar approach to the brute force attack using some request. This is achieved by sending requests over UDP to destinations. In this case the SIP proxy assumes that the requests were lost and will retransmit them in for the duration of TimerB, which has a default value of 32 seconds (Rosenberg *et al*. 2002b). After the expiry of TimerB, the proxy sends back a *408 Request Timeout response*. The attacker will not bother to send an ACK request back, the proxy will retransmit the 408 response for the duration of TimerH, which is by default also 32 seconds. This means that, for each transaction created by the attacker, the proxy will keep the transaction state for the duration of TimerB + TimerH, which taking the default values in (Rosenberg *et al*. 2002b), equals 64 seconds. Figure depicts the message flow that can be observed in such an attack.

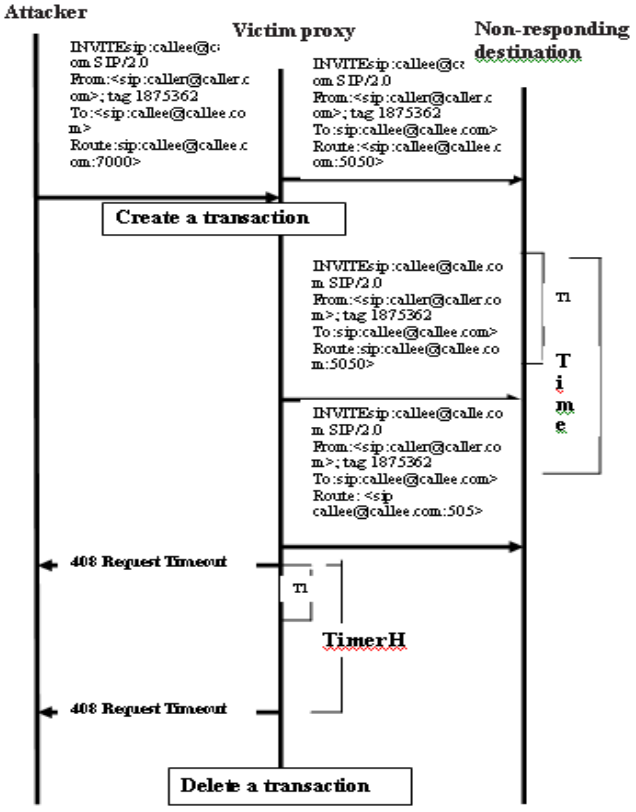

Fig 2: Incomplete transaction DoS attack

This kind of attack can be launched in different ways:

· Send requests to one or more UA that are registered , but that do not reply to the requests. It is possible if the receivers have no connection to the Internet, have been manipulated or the registered IP addresses are not allocated to any host.

· Another approach is for the attacker to include an IP address of a host in the *Request-URI* (name@IP address) instead of using of (name@domain name). If there is no SIP implementation at this IP address, then no SIP responses will be generated.

· If a request with a Route header then a proxy forwards a request based on the content of the *Route* header and not the Request-URI. In this case the attacker can indicate an IP address in the *Route* header request that belongs to a host without any reply.

III. POSSIBLE COUNTERMEASURES AGAINST ATTACKS

A. Countermeasures against memory exploitation attacks

Monitoring and filtering: Similar to web and mail servers SIP proxies need to maintain lists of suspicious users and deny those users from establishing sessions. It can be established by monitoring the transactions served by the proxy and logging user behavior, e.g. users that cause a sudden increase in the number of served transactions or users involved in broken transactions.
Authentication: Generally verifying the identity of a user before forwarding his messages would prevent spiteful behavior as the user would be easily traceable – naturally, this is only possible when an attacker to guess the identity of a valid user. Like HTTP, SIP uses digest authentication, which requires state maintenance at the server by storing the issued challenge. This can be misused for a broken session attack, if attackers ignore or incorrectly respond to authentication requests and start another session instead.

A solution to this problem could have been the usage of predictive nonces [139] that allow for stateless authentication and introduce limited message integrity. The construct is based on nonces being calculated in such a way which makes them valid only for validated messages within a time-window. When a challenge-response pair arrives at a server, the nonce is first verified to be correct, followed by the verification of the response. It works without any changes to the protocol. Stateless proxy: There is a certain computational expenditure associated with each SIP transaction at each proxy server and this cost is much greater for stateful proxy operation. Thus, an obvious protective measure for reducing the risk of memory exhaustion attacks is to perform as much of the server's functionality in stateless mode before going stateful. The "stateless barrier" should be used to perform as many security checks as possible – these may include

• Stateless authentication of users, forgoing the normal re-transmission algorithm. (By re-transmitting the 401 (unauthorized) or 407 (proxy authentication required) status response it amplifies the problem of an attacker using a falsified header field value, like Via, to direct traffic to a third party.
• Checks of unauthorized 3-rd party registrations,
• Detection of replay attacks,
• Presence of virus bodies,
• Filtering of well-known spam sources.

This functionality may be located in a separate server fronting stateful servers (load distribution). The other alternative is to have the stateless logic executed in the same server but to proceed to stateful execution only after all stateless checks succeed. For this purpose, the concept of a stateless UAS can be utilized. In this case, a server generates a reply to a request, sends it out and immediately forgets the request, its computational result and resulting reply. Request retransmissions are processed as new requests. After all the stateless checks, transactional state may be established. Some functions such as static forwarding (e.g. least-cost gateway routing) may be easily executed statelessly. The following list of functionality which inherently requires stateful mode:

• Request forking to avoid confusion of upstream clients uninformed of forking.
• Accounting to report on the results of transactions as opposed to reporting on all individual messages.
• Re-transmission buffer, particularly important if a SIP path is known to include a wireless hop to absorb too rush re-transmissions.
• Some services such as "forward on event" forward to voice mail on busy. In this case the original request needs to be kept for the new request instantiation. Servers using a registration database as the basis for the routing decision need to be at least conditionally stateful to preserve forwarding coherency throughout a session and avoid thereby routing irregularity due to REGISTER updates.

B. Countermeasures against CPU attacks

1) Server design:

The first line of defense against any DoS attack is achieved by using well-dimensioned hardware with fast CPUs and large memory and high speed network connections. The software itself needs to be designed with security, speed and attack possibility. This implies deploying some or all of the following server design options:

2) Clean and efficient implementation:

Implementers need especially to use efficient and fast memory allocation schemes, event handling and parsing mechanisms.

3) Parallel processing:

To avoid blocking incoming messages while the server is busy processing a message or waiting for an answer from an external server (e.g. AAA). A SIP proxy should be implemented using threads or parallel processes with each process or thread responsible for processing one message at a time. Here a core part only acts as a message scheduler distributing incoming messages between the processes. Each process is then responsible for parsing the message, initiating any DNS requests or requesting the execution of an application and finally forwarding the message. Note however, that even with a thread based server, server blocking is still a danger. If an attacker generates a higher number of

messages that cause the server to block than the available number of threads the server will still block.

IV. ATTACK AMPLIFICATION

SIP provides excellent means for attack amplification. Sip has ability to loop and fork requests may account for exhausting a CPU and server's memory with a single attack message. In loop amplification, the attacker needs to convince a proxy server to rewrite a request to a location, which resolves to a server itself. Using a header name Max Forward to moderate infinite loop, which is decremented for each traversed proxy.

The other amplification mechanism is forking. An attacker can easily register N locations with an address resulting in N-times higher overhead during every request sent to the address. That is, we consider an attacker having registered N + 1 accounts at N + 1 providers. At each provider the attacker registers N contact addresses pointing to the other accounts. With such a scenario a request would make each of the SIP servers at each of the providers to be involved in the routing of $N^{Max-Forwards}$ messages. In addition to the overload due to message processing, using forking to amplify an attack is especially attractive, as forking proxies need to be stateful. Thereby, this attack combines both CPU and memory exhaustion attacks.

To reduce the effects of this attack, the proxy might limit the value of Max-Forwards. That is the proxy might set a maximum limit for the value Max-Forwards that it might accept. If a request was received with Max-Forwards set to a higher value, the proxy would reset this value to its defined limit.

V. CONCLUSION

In this section we presented an overview of possible approaches for mounting denial of service attacks. We also distinguish between attacks and authentication protocols used by SIP but not part of SIP and attacks on SIP itself. The conclusion of the work is that SIP provides a large range of features that can be used to mounting DoS attacks. Denial of service attacks is a hard class of problem to solve. The inbuilt problem root is the desire to services to the public Internet together with the difficulty to recognize "friends" from "enemies".

REFERENCES

[1] Jeffrey Albers, Bradley Hahn, Shawn McGann, et al: An Analysis of Security Threats and Tools in SIP-Based VoIP Systems [EB/OL] September,2005.

[2] Text Available at: www.colorado.edu/policylab/Papers/ Univ_Colorado_VoIIP_Vulner.pdf.

[3] Rosenberg J, Schulzrinne H, Camarillo G et al. SIP: session initiation protocol [EB/OL]. June 2002.

[4] Y. Rebahi, D. Sisalem: SIP Service Providers and the Spam Problem [EB/OL] April 2005.

[5] Rosenberg: "Request Header Integrity in SIP and HTTP Digest using Predictive Nonces", expired Internet Draft, work in progress, IETF, June 2001. draft-rosenberg-sip-httppnonce-00.txt.

[6] Jiri Kutha, "Comparison of Service Creation Approaches for SIP", International SIP conference 2000, March 2000.

[7] de Laat, G. Gross, L. Gommans, J. Vollbrecht, D. Spence, "Generic AAA Architecture", RFC 2903 , Experimental, August 2000.

[8] 3GPP Technical Specification 3GPP TS 24.228 " Technical Specification Group Core Network; Signaling flows for the IP multimedia call control based on SIP and SDP", 3rd Generation Partnership Project, 2003.

[9] Text Available at: http://www.snocer.org/Paper/.

[10] Joachim Posegga, Jan Seedorf: Voice Over IP:Unsafe at any Bandwidth [EB/OL]. April 2005.

[11] Text Available at: http://www.informatik.unihamburg.de /SVS/papers/ Eurescom_VoIP-unsafe.pdf.

[12] Mark Collier, "Basic Vulnerability Issues for SIP Security," SecureLogix Corporation, March 2005.

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

[14] Nahum et al., "Evaluating SIP Proxy Server Performance," in Proc. NOSSDAV '07, 2007.

[15] Jens Fiedler, Tomas Magedanz, and Dorgham Sisalem. VoIP defender: highly scalable sip-based security architecture. In IPTComm '07: Proceedings of the 1st international conference on principles, systems and applications of IP telecommunications, page 11-17, New York, NY , USA, 2007. ACM.

[16] J. Peterson and C. Jennings. Enhancements for authenticated identity management in the session initiation protocol (SIP). Material available online at http://tools.ietf.org/html//rfc4474