A Framework for Detecting Malformed Messages in SIP Networks

Dimitris Geneiatakis, Georgios Kambourakis, Tasos Dagiuklas,
Costas Lambrinoudakis and Stefanos Gritzalis

Laboratory of Information and Communication Systems Security
Department of Information and Communication Systems Engineering
University of the Aegean, Karlovassi, GR-83200 Samos, Greece
Tel:+30-22730-82247
Fax: +30-22730-82009
Email:{dgen,gkamb,ntan,clam,sgritz}@aegean.gr

Abstract— **Internet telephony like any other Internet service suffers from security flaws caused by various implementation errors (e.g. in end-users terminals, protocols, operating systems, hardware, etc). These implementation problems usually lead VoIP subsystems (e.g. SIP servers) to various unstable operations whenever trying to process a message not conforming to the underlying standards. As Internet telephony becomes more and more popular, attackers will attempt to exhaustively "test" implementations' robustness, transmitting various types of malformed messages to them. Since it is almost infeasible to avoid or predict every potential error caused during the developing process of these subsystems, it is necessary to specify an appropriate and robust, from the security point of view, framework that will facilitate the successful detection and handling of any kind of malformed messages aiming to destruct the provided service. In this paper, we adequately present malformed message attacks against SIP network servers and/or SIP end-user terminals and we propose a new detection "framework" of prototyped attacks' signatures that can assist the detection procedure and provide effective defence against this category of attacks.**

I. Introduction

It is well known, that both protocol implementations and network applications are not fully conformant with the underlying standards or that they contain development errors in their source code, which might "pollute" a network with incorrectly formed packets. A number of common TCP implementation problems are already documented in [1]. Thus, an attacker may employ malformed messages in order to cause "unstable operations" to the computing system. A malformed message is any kind of invalid or non-standard message, skillfully formed by the attacker in order to exploit and eventually take advantage of, any implementation gap or dysfunction might exist in the target system.

Specifically for Internet applications or/and services, numerous distinct types of malformed message attacks have been already launched [2],[3]. Clearly, like any other Internet application or service, this problem cannot be avoided in Internet Telephony-Voice over IP (VoIP) implementations as well. Some research work that reveals security flaws caused by malformed messages in signaling protocols (such as H.323 and Session Initiation Protocol (SIP) implementations, have been already published in [4]-[6].

Moreover, attackers will keep trying to compromise the systems by utilizing properly adapted malformed messages. Malformed messages are characterized as a *high-level* type of attack that covers illegally formatted input. This security problem is often poorly understood and requires more research effort in order to be able to effectively protect implementations from this kind of attack. The PROTOS project [7] has made great strides to identify certain subclasses of malformed input. Processing malformed messages in VoIP networks can surprisingly give access to an unauthorized user or drive the provided service to various unstable operations and consequently cause Denial of Service (DoS). As a final point, the aforementioned issues implicitly affect the reliability and availability of VoIP service itself.

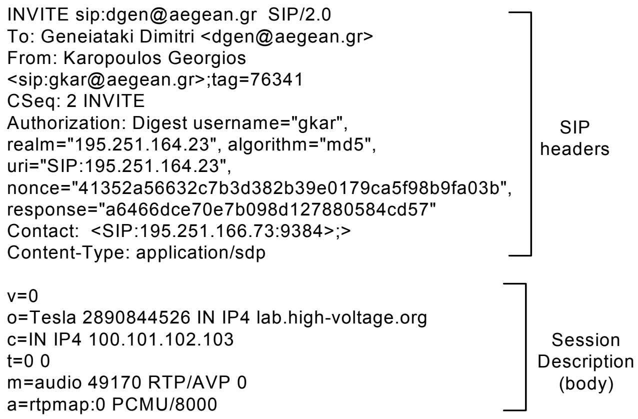
This paper aims to describe malformed message attacks against SIP network servers or SIP end user terminal, proposing a framework, consisting of prototyped attacks' signatures, that can assist to the identification and handling of such attacks. The rest of the paper is organized as follows: Section II describes special malformed messages that can be constructed in a SIP implementation. Section III briefly describes the procedure that an attacker can follow for launching a malformed message attack, while Section IV presents specific mechanisms for identifying and handling such attacks. Section V concludes the paper and provides pointers to future work.

II. SIP Malformed Messages

SIP is an application-layer signalling protocol for creating, modifying, and terminating multimedia sessions between one or more participants [8]. A SIP message can be either a request or an acknowledgment to a corresponding request, consisting of the header fields and the message body. The overall structure of a typical SIP message is depicted in Figure 1. SIP messages are text-based and are very similar to the HTTP format. According to RFC 3261 [8] all SIP stacks must be capable of implementing and processing the following standard SIP methods - messages: (a) REGISTER, (b) INVITE, (c) ACK, (d) CANCEL, (e) BYE and (f) OPTIONS.

The HTTP-like ASCII presentation of the SIP messages may initially be more attractive to attackers for vulnerability assessment than the rival signalling protocols (e.g. H.323, MGCP, SKINNY) with complex encodings. As a result, a malicious user can take advantage of any of the aforementioned SIP method - messages to mount this attack against SIP targets, which can be end-users' terminals or SIP Proxy Servers. Apart from the standard SIP methods/messages, there are also SIP extensions [9]-[11] for various SIP methods providing several complementary services that can be possibly utilized by potential attackers.

SIP subsystems have been designed and developed for processing messages that are valid and conformant with the SIP protocol syntax, as described in RFC 3261 [8]. An example of a valid and typical INVITE message that the SIP protocol syntax must be able to generate and process successfully is depicted in Figure 1.

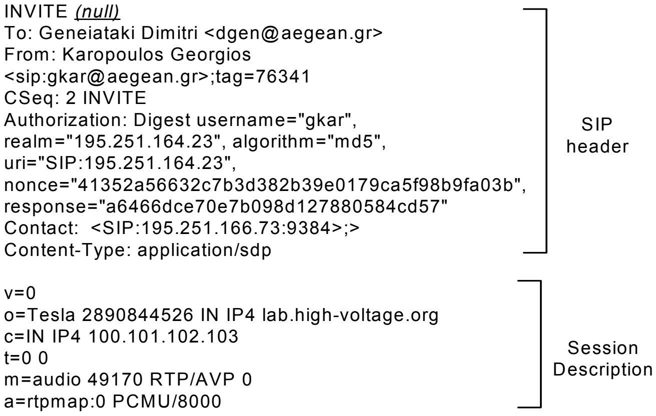


```
INVITE sip:dgen@aegean.gr  SIP/2.0
To: Geneiataki Dimitri <dgen@aegean.gr>
From: Karopoulos Georgios
<sip:gkar@aegean.gr>;tag=76341
CSeq: 2 INVITE
Authorization: Digest username="gkar",
realm="195.251.164.23", algorithm="md5",
uri="SIP:195.251.164.23",
nonce="41352a56632c7b3d382b39e0179ca5f98b9fa03b",
response="a6466dce70e7b098d127880584cd57"
Contact:  <SIP:195.251.166.73:9384>;>
Content-Type: application/sdp

v=0
o=Tesla 2890844526 IN IP4 lab.high-voltage.org
c=IN IP4 100.101.102.103
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Figure 1. Well formed typical INVITE message

It is highly likely that the attacker will try various malformed message combinations to discover a security problem/flaw towards the SIP-victim subsystem. For example, the INVITE message depicted in Figure 2 is invalid and cannot be generated by the standard SIP protocol syntax, due to the lack of a REQUEST-URI, which must follow the INVITE method [8].



```
INVITE (null)
To: Geneiataki Dimitri <dgen@aegean.gr>
From: Karopoulos Georgios
<sip:gkar@aegean.gr>;tag=76341
CSeq: 2 INVITE
Authorization: Digest username="gkar",
realm="195.251.164.23", algorithm="md5",
uri="SIP:195.251.164.23",
nonce="41352a56632c7b3d382b39e0179ca5f98b9fa03b",
response="a6466dce70e7b098d127880584cd57"
Contact:  <SIP:195.251.166.73:9384>;>
Content-Type: application/sdp

v=0
o=Tesla 2890844526 IN IP4 lab.high-voltage.org
c=IN IP4 100.101.102.103
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Figure 2 Example of Malformed INVITE message

Any message that either does not conform to or violate SIP's protocol specifications can cause security flaws in any SIP subsystem, but usually, it is very difficult to distinguish between all possible legal and illegal messages. In a nutshell, possibly there are inputs that might not have been considered properly when implementing the SIP stack installed in each SIP product.

III. Mounting the Attack

Normally, the attacker does not have a standard method for launching an attack. Therefore, in a sense, the behaviour of an attack is unpredictable. This is also true for SIP malformed message type attacks. For example, the attacker may construct malformed messages utilizing a "brute force" attack method, exhaustively trying all possible SIP message combinations. Alternatively the attacker can follow a more general procedure, which could be expressed in the following, repeatedly executed, algorithmic steps:

1. Discover the target's SIP capabilities.
2. Construct the malformed message.
3. Test the derived "crafty" message against the SIP target.

The main "advantage" of such an approach is that the assault cannot be easily identified in its prime stages, as the defence mechanisms in place are not usually able to promptly detect it.

A. Discovering the target's SIP "capabilities"

The initial step before an attacker launches a malformed message attack, is to discover the SIP "capabilities" of a particular SIP target/subsystem.

It is known, that REGISTER message and OPTIONS response can give information about any SIP User Agent's (UA) capabilities. This sensitive information is included in *Contact* header in REGISTER message and *Allow* header in response of the OPTIONS request. In every case, these messages can be utilized from the attacker in two different ways aiming to discover the User Agent's (UA's) capabilities.

In the first one, the attacker can simply sniff SIP packets (especially SIP REGISTER packets) while a registration to a SIP registrar server is taking place. The other one merely utilizes the OPTIONS message. Figure 3 depicts the message flow for this method. Under these circumstances, the attacker creates an OPTIONS message, which is sent to the SIP victim. The target responds to the OPTIONS message, so the attacker discovers the SIP target's implemented methods/messages (capabilities). For instance, the returned capabilities may reveal, among other things, the vendor and the version of the potential SIP target product, which in turn, expose existing vulnerabilities.

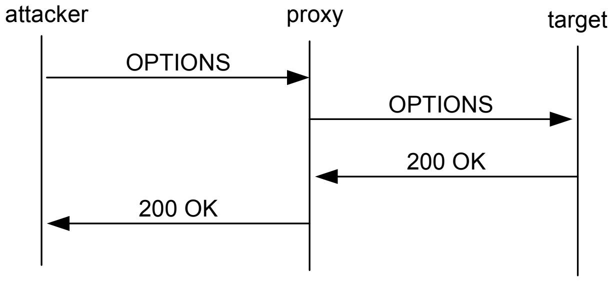


Figure 3 Discovering User Agent Server "Capabilities"

B. *Construction of a malformed message*

The second step towards launching the attack is to construct an appropriate malformed message. Figure 2 presents the structure of a malformed INVITE that we were able to generate. A list of such INVITE malformed messages also exist in the test-suite PROTOS [7]. Moreover, it is very likely for the attacker to send non-implemented (invalid or non-standard) messages provoking eventually the target to crash. Besides that, as already mentioned in Section II, other implemented messages like REGISTER, BYE, CANCEL, and other non-standard messages can be also employed to have this attack compiled much in the same way.

C. *Restrictions-Limitations and Possibilities for the attack*

Any malicious user who is located (or not) in the same domain with the victim can launch this attack. There is no real restriction for the attacker to prevent him from launching a malformed message attack, in absence of any underlying security mechanism that protects message integrity, confidentiality and origin authentication. Even though, the existing underlying SIP security mechanisms (e.g. Secure Socket Layer-SSL, IP Secure-IPsec, Secure MIME-S/MIME) are only able to protect against "outsiders" and not against "insiders", who are normally the legitimate users. However, considering this situation, an outsider will endeavour to employ his SIP proxy in order to amplify the DoS effects of specially fabricated malformed, invalid or non-standard SIP messages towards the corresponding SIP target.

IV. Defending Against Malformed messages

To facilitate the development of a robust and secure VoIP service, highly immune to malformed message attacks, one has to employ a number of prevention and detection mechanisms. Having these mechanisms acting simultaneously, it is possible to create a more secure environment.

A. *Countermeasures and remedies*

Input validation procedures must be considered vital for the security of VoIP services. The lack of any validation in data input process is responsible for security flaws caused by malformed messages. The employment of gateways to filter malicious input at the Internet application level has also been studied [12]. Current firewall technologies incorporate packet inspection [13] for validating input data. The same techniques can be applied in SIP architectures using the Middlebox Communication approach [14].

Moreover, the utilization of underlying security mechanisms (e.g. SSL, IPsec, S/MIME) according to RFC 3261 can substantially restrict or prevent the origination of malformed messages. However, as already mentioned in Section III.C, it is always possible for an attacker to utilize another SIP proxy to amplify the hazardous effects of the malformed messages. Additionally, these mechanisms do not provide any real security against internal-authorized (malicious) users.

Another possible countermeasure that can restrain this attack is the authentication of the OPTIONS messages. Additionally, the utilization of underlying security mechanisms is considered mandatory to protect the confidentiality of the REGISTER and OPTIONS messages against eavesdroppers. The employment of these countermeasures does not mean that the aggressor cannot launch the attack, but things become more difficult for him.

B. *Detection "framework"*

No matter how strong the existing security prevention mechanisms in VoIP Services are, there is always the possibility for a malicious user to manage to by-pass them. So, in case an internal user launches an intrusion attack, it is quite probable that none of the existing prevention mechanisms will trigger an alarm. For example, considering a legitimate SIP user who generates a malformed SIP message and then signs it with his private key. There is no doubt that this attack can be hardly defeated by the usual prevention mechanisms and awake the existing countermeasures.

To avoid such situations, the employment of an Intrusion Detection System (IDS) for the provided VoIP services is considered mandatory. On the other hand, in some cases, it is more economical to prevent only the uppermost attacks and detect the rest, than trying to prevent everything in a much higher cost. In addition, a detection system can be considered quite sufficient for protecting VoIP against malformed message attacks. In these systems, any distinct attack is described through some specific static structure, known as the attack's "*signature*".

Malformed message attack in SIP architectures can be similarly confronted by identifying, categorizing and prototyping the corresponding signatures. The proposed signatures are based on the SIP message syntax, which is fully specified in RFC 3261 [8]. Since all SIP messages are based on this syntax, it will be attainable to embed a light SIP IDS mechanism in a slightly modified SIP protocol stack. The signatures developed are marked out mainly for the most utilized SIP messages that current SIP User Agents implement.

The detection signatures are based in the structure depicted in Figure 4. Each signature is composed by the identified malformed message (SIP-MESSAGES) optionally followed by some additional rules.

```
SIP-MESSAGE
(based in SIP-GRAMMAR)
 additionall rules
```

Figure 4 General Form of a Signature

The basic idea is to construct a general identification-detection rule that can be easily applied to any SIP message, independently from the SIP-method (INVITE, REGISTER, BYE, etc) used.

Figure 5 presents the structure of this general rule. The first line represents the SIP Method, the URI and the corresponding header. This detection rule capitalizes on the fact that any SIP message must have a SIP method with the appropriate destination address followed by one or more message headers. Note, that not all SIP messages are mandate to have a message body. Moreover, additional rules add an increased security level and can effectively characterize a message as malicious or not. For example in the depicted rule, both the SIP method and the message header are prohibited from having the NULL value.

```
SIP_METHOD SIP-URI | SIPS-URI MESSAGE HEADER+
[MESSAGE_BODY]

additionall rules
SIP_METHOD!=NULL
MESSAGE_HEADER!=NULL
size_of(SIP_METHOD)>%constant% e.g 50 bytes
size_of(MESSAGE_BODY)>%constant%
```

Figure 5. General Detection Rule

However, there are cases (some very well known malicious messages) that cannot be identified by this generally structured rule. Under these circumstances (exceptions), special rules must be formed for each distinct SIP-method. For example, INVITEs which do not have a specific header (e.g Content-Type, Call-ID) are characterized as invalid. Figure 6 describes a detection signature framework for INVITE messages. Note, that this detection signature is very similar to a valid INVITE message. The main difference is that the message is characterized as "malicious" when any of the mandatory message headers is not in place or any of the additional rules triggers it. For instance, concerning this signature, there are two additional rules, which restrict the value of the Content-Length header. This value must be greater than zero and equal to the size of the MESSAGE_BODY expressed in bytes. If any of these rules is not satisfied or any mandatory header is missing, then the message must be discarded, perhaps giving some feed to the IDS too.

```
INVITE_METHOD SIP-URI | SIPS-URI MESSAGE HEADER+
MESSAGE HEADER =Via | Max-Forwards | From* |To* | Call-Id*
                    |CSeq* | Contact* |User-agent
                    |Authorization |Event |Content-Length*
                    |Content-type*|Record-Route
INVITE_METHOD="INVITE" | %x49.4E.56.49.54.45
MESSAGE_BODY
additionall rules
%Content-Length% >0
%Content-Length%==size_of(MESSAGE_BODY)
(*)mandatory fields
```

Figure 6. Detection Signature for INVITE messages

Another reason that prevents the employment of general structured rules only, is that different SIP methods require different message headers. Thus, the combination of general and special targeted rules can establish a robust identification framework to protect from SIP malformed messages. In addition, the administrators of each domain are responsible to utilize the appropriate rules (what is permitted and what is not) depending on the security policy determined beforehand.

V. Conclusions and Future Work

Availability, reliability and security in services like VoIP are critical and thus they must be protected, at least to the same degree as in Public Switched Telephone Network (PSTN). Various errors, gaps or even oversights originated during the implementation phase of the VoIP signaling protocols, can be exploited by potential attackers to gain unauthorized access or cause a DoS to the offered VoIP service. One method that the attackers can employ to disturb normal operations and undermine the provided service is the construction of malformed messages.

In this paper, we describe how an attacker can launch a malformed message attack against SIP subsystems in a VoIP network. A new signature-based detection "framework" is provided that is capable of identifying malformed messages in SIP networks. The proposed framework can be easily embedded to any standard SIP stack, and furthermore, co-operate with existing IDS systems. Another realizable possibility is to incorporate a light IDS to the SIP protocol stack itself.

However, the overheads, in terms of performance, introduced in SIP as a result of the proposed solutions are still under inspection. Besides that, we esteem that a slight modification of this aggression can also be applied in any VoIP service, independently from the underlying signaling protocol used. The accomplishment of this goal, currently under inquiry, will contribute a great deal in VoIP security, availability and reliability.

Acknowledgment

This work has been performed in the framework of the IST-2004-005892 project SNOCER, which is funded by the European Union.

REFERENCES

[1] Paxson V., Auman M., Dawson S., Fenner W., Griner J., Heavens J., Labey K., Semke J., and B.Volt,. "Known TCP implementation problems". RFC 2525, March 1999.

[2] CERT-In Advisory CIAD-2003-09 "Buffer Overrun In RPC Interface Could Allow Code Execution and Denial of Service" August 2003.

[3] Fontana J., "Exchange Server 5.5 Bug Could Be Exploited for Attacks", http://www.pcworld.com/resource/article/0,aid,33882,00.asp , November 2000.

[4] "Asterisk SIP Implementation Issue", http://www.atstake.com/research/advisories/2003/a090403-1.txt, August 2003

[5] CERT® Advisory CA-2004-01, "Multiple H.323 Message Vulnerabilities", http://www.cert.org/advisories/CA-2004-01.html, April 2004..

[6] CERT® Advisory CA-2003-06, "Multiple vulnerabilities in implementations of the Session Initiation Protocol (SIP)", http://www.cert.org/advisories/CA-2003-06.html February 2003.

[7] Wieser C, Laakso M, Schulzrinne H , "Security testing of SIP implementations", http://compose.labri.fr/documentation/sip/Documentation/Papers/Security/Papers/462.pdf, 2003.

[8] Rosenberg J., Schulzrinne H., Camarillo G., Johnston A., Peterson J., Spark R., Handley M., Schooler E., "Session Initiation Protocol", RFC 3261, June 2002.

[9] Donovan S. , "The SIP INFO Method", RFC 2976, October 2000.

[10] Rosenberg J., "The Session Initiation Protocol (SIP) UPDATE Method", RFC 3311, September 2002.

[11] Spark R., "The Session Initiation Protocol (SIP) Refer Method", RFC 3515, April 2003.

[12] Scott D. and Sharp R., "Abstracting Application-Level Web Security," Proc. 11th Int'l World Wide Web Conf., ACM Press, New York, May 2002, pp. 396-407.

[13] Dharmapurikar S., Krishnamurthy P., Sproull T., and Lockwood J., "Deep Packet Inspection Using Parallel Bloom Filters." In Proceedings 11th Symposium of High Performance Interconnects (HOTI'03), pages 44-71, 2003.

[14] Srisuresh P., Kuthan J., Rosenberg J., Molitor A. and Rayan A: "Middlebox Communication Architecture and framework", IETF, RFC 3303, August 2002.