Detecting More SIP Attacks on VoIP Services by Combining Rule Matching and State Transition Models *

Dongwon Seo, Heejo Lee, and Ejovi Nuwere

Abstract The Session Initiation Protocol (SIP) has been used widely for Voice over IP (VoIP) service because of its potential advantages, economical efficiency and call setup simplicity. However, SIP-based VoIP service basically has two main security issues, malformed SIP message attack and SIP flooding attack. In this paper, we propose a novel mechanism for SIP-based VoIP system utilizing rule matching algorithm and state transition models. It detects not only two main attacks, but also covers more SIP attacks. Instead of simply combining rule comparison and counting number of SIP messages, we develop secure RFC 3261 rules based on existing RFC 3261 rules, so that proposed mechanism shows 26% higher detection rate for malformed attack. Moreover, we utilize session information and define the features of each state in order to detect abnormal situations including SIP flooding. As the result, it is shown that the proposed mechanism provides not only higher accuracy, but also covering more SIP attacks including two main attacks.

1 Introduction

Telephone is definitely an important communication tool. As the Internet is being popular, Voice over IP (VoIP), also called Internet telephony, has become a promising communication medium owing to its economical rates and additional features such as video conversation, SMS and messenger services. It also means that VoIP services are facing on known and unknown security threats. As shown in several studies on VoIP security [7, 15, 5], there are lots of security problems in VoIP services. Actually, there are some existing tools to verify vulnerabilities of VoIP soft-

D. Seo and H. Lee are with Korea University, Seoul 136-713, Korea, and E. Nuwere is with SecurityLab Technologies, e-mail: {aerosmiz, heejo}@korea.ac.kr, ejovi@ejovi.net.

* This work was supported in part by the ITRC program of the Korea Ministry of Knowledge Economy.

Please use the following format when citing this chapter:

Seo, D., Lee, H. and Nuwere, E., 2008, in IFIP International Federation for Information Processing, Volume 278; *Proceedings of the IFIP TC 11 23rd International Information Security Conference*; Sushil Jajodia, Pierangela Samarati, Stelvio Cimato; (Boston: Springer), pp. 397–411.

wares. However, most of them simply scan known vulnerabilities and produce a report. For more robust VoIP services, it is necessary to design a mechanism which is capable of detecting specific suspicious packets and attack conditions without interrupting existing VoIP services.

There are two VoIP session protocols, SIP and H.323. However, SIP is recently being chosen because its simpler connection process and easier implementation for the Internet [9]. Therefore, we focus on the security issues of SIP-based VoIP services. Nonetheless, the principles of our study can be applicable to H.323 VoIP services.

Technically, SIP-based VoIP services consist of two different protocols, SIP and RTP (Real-time Transport Protocol). SIP is a signaling protocol to establish and terminate sessions. On the other hand, RTP is a media protocol to transfer multimedia data. Thus, there are two categories of attack along with the two protocols. One is SIP related attacks, which cause unexpected results such as service malfunction, session connection between wrong users, and incorrect billing to wrong users. Another is RTP related attacks, which include voice eavesdropping and media spamming. In exploring the questions of both SIP and RTP attacks, we first consider SIP attacks due to their growing impacts on VoIP services.

SIP protection is very important in the sense that SIP is in charge of session initiation, connection and termination. Especially, SIP is susceptible to two types of attacks, malformed message attacks and SIP flooding attacks. It is easy to forge the header fields of a SIP message since the message is based on plain text. And there are many tools to generate SIP packets for launching SIP flooding attacks. However, previous works do not consider both attacks simultaneously, but detect only one type of attacks at a time, either malformed messages [3] or flooding attacks [1].

Main contributions of this study are twofold.

1. Unlike existing researches which detect two main SIP attacks (malformed and flooding attacks) separately, we develop a new approach by combining rule matching and state transition models, and it detects not only two main attacks, but also covers three more SIP attacks as utilizing SIP features with affordable overhead.
2. Because of plain text-based SIP message, it is difficult to cover all variant malformed messages which can exploit vulnerabilities of SIP-based VoIP services such as buffer overflow and string format exception. Especially, there is no research that provides statistical experiment for detecting malformed SIP messages so far. Therefore, we develop secure RFC 3261 rules using regular expression based on RFC 3261 ABNF rules. As a result, from the experiment based on 2426 malformed cases of PROTOS test suite, our proposed approach shows 26% higher detection rate than using original RFC 3261 rules.

The rest of this paper is organized as follows. In Sect. 2, we introduce related works. Threat models for SIP and RTP are discussed in Sect. 3. And, we propose a novel mechanism for detecting more SIP attacks in Sect. 4. The evaluation of the proposed mechanism is shown in Sect. 5. Finally, we summarize our result and conclude the paper in Sect. 6.

2 Related Work

There exist some researches using state machines for intrusion detection. One of them is State Transition Analysis Technique(STAT) [6], which is a rule-based intrusion detection approach. STAT is a general method that recognizes computer penetrations easily using rule-based state diagram. There are different versions of STAT. NetSTAT [12] is to determine which network event should be monitored, and WebSTAT [13] is to detect malicious behaviors for web servers according to analyzing web requests.

In addition, Snort is the most broadly deployed IDS around the world and it has many attack patterns, over 6000. To protect VoIP system, it may be possible to apply to an existing IDS. However, there are some problems when we use a current IDS directly to protect VoIP system [16]. First, VoIP service is based on session while IDS detects attacks based on packets. It means that IDS monitors every single packet and compares it with pre-defined rules, but it is necessary for a VoIP service to distinguish which session the packet belongs to. Second, although Snort provides stateful detection for TCP-based protocols like HTTP and FTP, it does not help in processing stateful VoIP sessions. Finally, VoIP service is formed combining of multi-protocol, such as the signaling protocol SIP and the media protocol RTP. If an attack is performed across protocols, conventional IDSs fail to detect it. Therefore, we need to develop intrusion detection technologies dedicated to VoIP services.

Several studies have been done for protecting VoIP services. SCIDIVE by Yu-Sung Wu [16] is an architecture which provides stateful and cross protocol detection. SCIDIVE is able to detect attacks in both protocol, SIP and RTP. To examine SIP format, SCIDIVE uses rule sets including standard SIP rules. However, there are many malformed SIP messages which is formed as standard but dangerous. For example, %s%d%caaa.com follows a standard form, even though it may be dangerous because of format string like %s%d.

Hemant Sengar also proposed a VoIP defense mechanism by the use of state machines [10]. The mechanism uses cross protocol state machines which define attack detection patterns. The mechanism also has an advantage of detecting across two protocols. However, it is not a flexible mechanism because it needs lots of state machines to protect against various attacks.

There is a similar approach to detect malformed SIP messages [3]. It proposes a framework based on the rules for valid SIP messages. The key idea is that normal SIP messages should have mandatory fields and fit to pre-defined byte size. Nonetheless, this mechanism allows to pass malformed SIP messages, which include the messages whose mandatory fields and byte sizes are even less than predefined ones. Considering that SIP header fields use plain text, we have to examine the content of each header that may contain abnormal string formats such as non-ASCII, malformed UTF-8 and escape characters, and so forth.

Eric Y. Chen proposed DoS detecting method on SIP systems [1]. It also utilizes RFC 3261 state transition models, and defines additional state and upper bounds for error conditions. One drawback of this approach is that malformed SIP messages are not considered properly. Although this mechanism is very effective to detect DoS

or flooding attacks, malformed SIP messages are definitely hazardous because they cause the malfunction of a VoIP service. In contrast to this approach, we propose a mechanism that is able to detect both malformed SIP messages and flooding attacks at the same time.

3 Threat Model

From the previous researches, [7]and [1], we could categorized VoIP attacks into six groups (three SIP related and three RTP related attacks) by their protocols and behaviors.

VoIP attacks can be divided into two categories: SIP attacks and RTP attacks. Since SIP takes significant roles of session initiation, connection and termination, we need to consider SIP attacks first. RTP attacks are briefly discussed in this Sect., and they are out of our scope. We do not consider all kinds of SIP attacks like the attacks derived from IP features such as spoofing attack. Our attention is directed to SIP attacks derived from SIP features such as malformed message and SIP flooding attacks [11]. These two attacks are strongly connected to SIP systems and exploit their vulnerabilities. In the light of this consideration, we propose a novel approach that is able to handle with those two attacks simultaneously.

Malformed Message Attack: This is one of the most representative case using the vulnerabilities of text-based protocol. Attackers are able to cause malfunctions of proxy server or UA by manipulating SIP headers. For instance, overflow-space, overflow-null, specific header deletion and using non-ASCII code are involved in malformed message attacks.

SIP Flooding Attack: IP phones generate requests or responses to send to a specific UA, called by `victim`. As a result, a single UA is overwhelmed by receiving excessive SIP messages within a short duration of time, so that the UA cannot provide normal services. INVITE flooding is one of the most typical attacks. Basically, flooding attack is also the issue of IP layer. In case of INVITE flooding, however, it could be more annoying attack for the VoIP user because the one should see many call requests and hear ringing.

Spoofing Attack: Two kinds of spoofing attacks are possible, IP spoofing attack and URI spoofing attack. IP spoofing attack is to forge IP source addresses in order to pretend a trusted user. And, IP spoofing is the intrinsic security problem in TCP/IP protocol suites and it is not in the scope of our study on VoIP security. URI spoofing attack is a particular case in malformed message attacks. The attacker who hijacked SIP messages between two UAs forges their URI field, so the attacker can hide himself from tracebacks. If spoofed BYE requests (BYE DoS attack) are sent to a victim, the call will be terminated by the attacker.

In addition to the SIP attacks, there are several kinds of RTP attacks. RTP attacks can be classified into three categories: RTP flooding attacks, media spamming attacks, and man-in-the-middle (MITM) attacks. RTP flooding attacks are similar

to SIP flooding attack, but they use RTP packets. Media spamming attacks, also known as SPIT (Spam over Internet Telephony), have been an annoying problem that disturbs a user who does not want to receive a call for advertisement. Finally, MITM attacks are similar to eavesdropping. It is one of the most critical issues in RTP attacks.

4 The Proposed Mechanism

In this Sect., we propose a new approach to detect SIP attacks including two main types of SIP attacks, malformed messages and flooding attacks.

4.1 Background

This part gives an overview of basic knowledge about the constitution of SIP message and how to call-setup and tear-down on SIP.

4.1.1 SIP Messages

A SIP message basically consists of two parts, message header and body. A message header contains essential user information such as URI (Uniform Resource Identifiers), method and Call-ID. A message body is described as SDP (Session Description Protocol) which are informed for media encoding scheme [4].

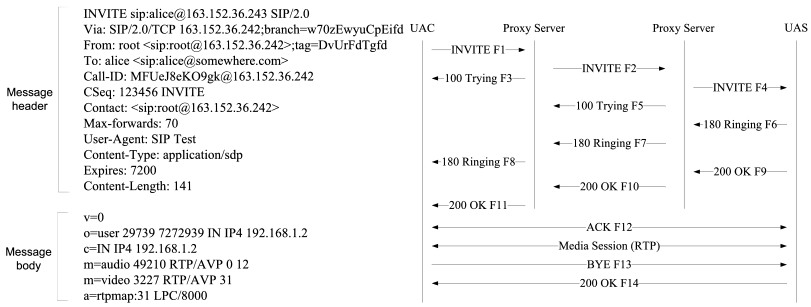

Fig. 1 Normal INVITE request (left) and SIP call-setup and tear-down process (right).

There are six general requests; INVITE, ACK, BYE, OPTIONS, REGISTER, and CANCEL. INVITE is for making a call to the other, ACK is corresponding request to response, BYE is to terminate a call, OPTIONS is for getting information such as user capability, REGISTER is for signing in or out from VoIP provider, and

CANCEL is to abort last request. Responses, which are three digit numbers, comprise six classified groups; Provisional, Success, Redirection, Client Error, Server Error, and Global Failure. Figure 1 (left) is an example of a normal INVITE request.

4.1.2 The Call-setup and Tear-down Process on SIP

In order to set up a call, UAC (User Agent Client, caller) sends an INVITE request to UAS (User Agent Server, callee). Proxy server forwards it to UAS and sends 100 Trying response to UAC. After the UAS receives INVITE request, it transfers 180 Ringing and 200 OK responses subsequently. Finally the UAC gets OK response, sends ACK request and the connection is established. Figure 1 (right) indicates such a process.

4.2 The Concept of the Proposed Mechanism

The VoIP service uses SIP when it makes call-setup and tear-down and takes RTP while transmitting media stream data. Since SIP is on the upper layer of IP layer, SIP also has weak points such as flooding Besides, text-based message header is always exposed to various text-modified attacks such as string overflow. To corresponding SIP attacks, we design a detection mechanism which consists of three parts: malformed SIP detection, session management, and state verification. The most significant modules are malformed SIP detection module that performs rule matching and header field categorization, and state verification module that is related to four state transition models [2]. Figure 2 is an overall flow chart of our mechanism.

4.3 Malformed SIP and Invalid Header Field Detection

Malformed SIP detection module covers two SIP attacks, malformed SIP and invalid header field attacks.

First of all, to apply RFC 3261 rule sets for real VoIP services, we convert RFC 3261 ABNF rules into regular expressions. Rule matching algorithm decides whether the header of a packet follows its standard forms. Malformed SIP packets including unmatched or undefined headers can be blocked or considered to pass.

There are over 280 rules in RFC 3261, and we can define the standard forms of the SIP messages in the rules. However, we found that the original RFC 3261 rules have some vulnerabilities to cover many kinds of malformed SIP messages. For instance, the regular expression corresponding to the `userinfo` rule in RFC

[2] INVITE server, INVITE client, Non-INVITE server, and Non-INVITE client transition models

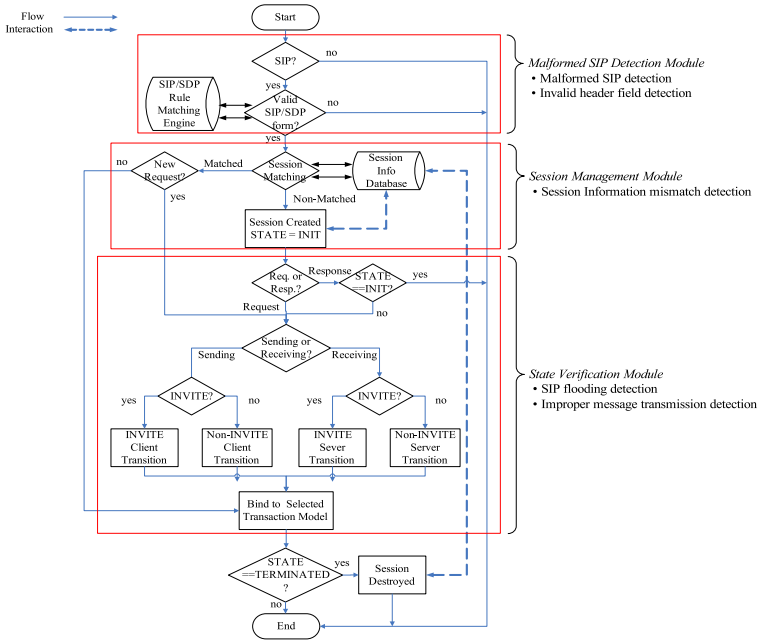

Fig. 2 Overall flowchart of proposed mechanism

3261 is like this, **userinfo:((#user#)(:#password#)?)**. What if an input is for extremely long user ID or password? It may cause unexpected result such as overflow exception. For one more simple example, there is an ABNF rule for port number:

port=1*DIGIT

The corresponding regular expression for the ABNF rule is

port=\d+

which means that a port should be a number more than one digit. Nonetheless, the rule does not check length of the port number causing overflow-integer. Thus, we change from the original rule to a secure one,

port=(\d{0,4}|[1-5]\d{4}|6[0-4]\d{3}|65[0-4]\d{2}|655[0-2]\d|6553[0-5])

because port number is from 0 to 65535. For instance, port number **65540** is mismatched by the port rule, **655[0-2]\d**. An adversary can make a lot of exceptional cases like the example, and they may cause malfunctions of SIP-based VoIP services. For that reason, we apply secure SIP rules that restrains size and format of string and number. Table 1 shows that some example of comparison between regular expressions based on RFC 3261 ABNF rules and secure regular expressions. For instance, `user` field allows only alphabet, number, '_', '-' and must not be over

twelve characters. Formalizing of SIP standard form is capable of recognizing not

Table 1 Rule comparison between original and secure regular expressions

RFC3261 regular expressions	Secure RFC3261 regular expressions
user:((#unreserved#|#escaped#|#user_unreserved#)+)	user:((***#alphanum#***|_|\-)***{1,12}***)
password:((#unreserved#|#escaped#|\&|\=|\+|\$|\,)*)	password:(((#unreserved#|#escaped#|\&|\=|\+|\$|\,)*)***{0,12}***)
SIP_Version:(SIP\/\d\.\d)	SIP_Version:((SIP\/\d\.\d)***{7,9}***)
extension_method:(#token#)	extension_method:(***#ASCII_NAME#{1,20}***)
protocol_version:(#token#)	protocol_version:(\d***{1,2}***\.\d***{1,2}***)
display_name:((#token##LWS#)*|#quoted_string#)	display_name:(***(\w){1,32}***|#quoted_string#)
callid:(#word#(\@#word#)?)	callid:(***#ASCII#{1,50}***(\@(***\w\. *){1,32}***)?)
Max-Forwards:(Max-Forwards#HCOLON#\d+#CRLF#)	Max-Forwards:(Max-Forwards#HCOLON#\d***{1,4}***#CRLF#)

only known malformed SIP packets, but also unknown ones. In addition, it is very flexible to being adapted reformed standard by adding or editing existing rules.

Moreover, categorizing mandatory and optional header fields for each SIP message in our secure RFC 3261 rule sets, it is possible to filter out suspicious SIP messages which is well-formed SIP but includes non-allowed header fields. For instance, SIP requests must contain `Call-ID`, `CSeq`, `From`, `Max-Forwards`, `To`, and `Via` header fields. Also, ACK should not contain `Subject` header field. Through these kinds of rule grouping, malformed SIP detection module performs stronger rule matching. Table 2 is a categorized table to detect invalid header field for ACK message.

Table 2 Categorized header fields table for ACK to detect invalid header field

Types	Header fields
Mandatory (6)	Call-ID, CSeq, From, Max-Forwards, To, Via
Optional (13)	Authorization, Contact, Content-Disposition, Content-Encoding, Content-Language, Content-Length, Content-Type, Date, MIME-Version, Record-Route, Route, Timestamp, User-Agent
Non-allowed (25)	Accept, Accept-Encoding, Accept-Language, Alert-Info, Allow, Authentication-Info, Call-Info, Error-Info, Expires, In-Reply-To, Min-Expires, Organization, Priority, Proxy-Authenticate, Proxy-Authorization, Proxy-Require, Reply-To, Require, Retry-After, Server, Subject, Supported, Unsupported, Warning, WWW-Authenticate

4.4 Flooding and Improper Message Transmission Detection

State verification module decides whether or not each SIP message is normal based on current state. We adopt four modified state transition models from RFC 3261,

and focus on `INVITE server transition model` to describe how it works in this paper. The dashed lines indicate an abnormal (either attack or suspicious) condition for each state.

Figure 3 describes `INVITE server transition model`. The model is selected when a host receives INVITE message. Each state compares number of messages with threshold in order to check flooding condition. Especially, in `Confirmed` state, receiving INVITE and all kinds of responses are identified as abnormal conditions. Like these, through state verification module, it is possible to detect flooding attack and improper message transmission. Figure 4 shows an example of improper message transmission. Bob is now on `Confirmed` state, which allows only ACK message. If Trudy sends INVITE message, however, we can detect it.

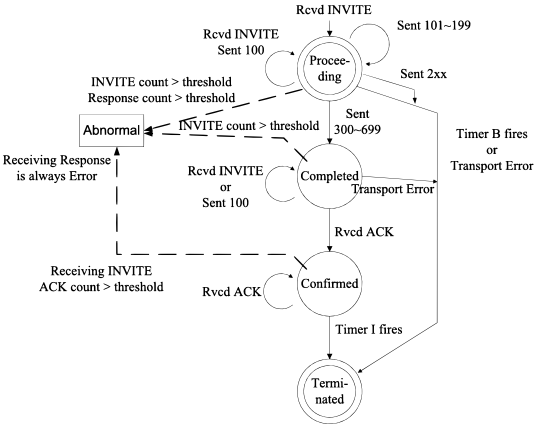

Fig. 3 INVITE server transition model: `Abnormal` state handles flooding condition and improper message transmission.

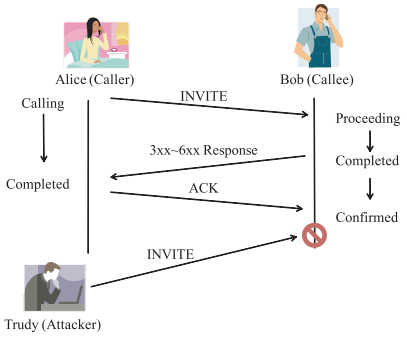

Fig. 4 Improper message transmission: Trudy sends INVITE message to Bob, which is unacceptable to Bob's current state, `Confirmed`.

4.5 Session Information Mismatch Detection

Session management module creates a new session after receiving or sending INVITE request, and destroys the session after receiving or sending BYE request. The followings are the information which should be stored in session management module.

- URI: to distinguish UAC and UAS.
- Selected state transition algorithms: the form of queue containing history of selected state transition algorithms.
- Current state: current state of most recent selected state transition algorithm.
- Error code: there are three levels, e.g. pass, warning and abnormal.
- `Sequence number`: 32-bit unsigned integer. A response copies the sequence number from received request, and it adds certain increment like 256 when sending a new request.
- `Call-ID`: it uniquely identifies a particular invitation or all registrations of a particular client.

Comparing current sequence number and Call-ID of each session with previous ones, we are able to detect session information mismatch. this module has a similar concept to stateful inspection.

5 Evaluation of the Proposed Mechanism

In order to measure the effectiveness of the proposed mechanism, we used publicly available attacking tools such as PROTOS [2] and SiVuS [14]. PROTOS is a popular VoIP vulnerability assessment tool and `PROTOS test-suite:c07-sip` provides a lot of malformed SIP messages. SiVuS is used for launching SIP flooding attacks by generating overwhelming SIP messages. The PROTOS suite has been widely used and publicly available to evaluate the implementation level security and robustness of Session Initiation Protocol (SIP) implementations. There are 4527 malformed SIP test cases. SiVuS is a free VoIP vulnerability scanner which has the ability to generate packets and SIP header fields can be edited by a user.

Moreover, we developed two application programs, namely `VoIPDefender` and `VoIPAttaker`. VoIPDefender is a prototype implementation of the proposed mechanism, and VoIPAttacker is a SIP attacking tool whose input is a file name for the PROTOS suite and generates attack patterns according to each test case.

At last, to verify whether our proposed mechanism disturbs existing VoIP services, five SIP softphones are chosen from ”myvoipprovider.com” web site [8], which offers top 100 raking of 155 international VoIP providers. The last comparison is updated on December 2007. We picked five softphones providing free PC to PC VoIP services based on SIP. The five softphones are Globe7, Vbuzzer, VoIPGo, Gizmo Project and SJPhone.

5.1 The Result for Malformed SIP Attacks

A subset of SIP from PROTOS suite, namely INVITE messages, was chosen as the subject protocol for vulnerability assessment through syntax testing and test-suite creation. An exceptional element is a piece of data designed to provoke undesired behavior of the test subject. An exceptional element can violate the protocol specification, but often it is legal or in the hazy region between legal and illegal constructs [2]. We could get 4527 test cases of malformed SIP packets, and 2426 cases

Table 3 SIP exceptional cases in PROTOS test suite

# Case	Exceptional Elements	Description
1	Overflow-general, space and null; Format string; UTF-8; ANSI-escape	Repetition of general character, space or null; Using format string. Ex) %s%d%f; UTF-8 code. Ex) Chinese characters; Start with characters ESC (ASCII 27d / 1Bh / 033o) and [(left bracket).
2	SIP-URI	Invalid SIP-URI form. Ex) sip: aaa:bbb@ccc.ddd, port number should be a number not character like “bbb.”
3	SIP-Version	“SIP/” must have existed. Ex) SIP:2.0
4	IPv4-ASCII	The number range should be from 0 to 255.
5	Integer-ASCII	Number ranges are needed. Ex) port number
6	Overflow-colon	Only one colon is allowed. Ex) sip::::::::invalid.com
7	SIP-tag	Only one semi-colon is allowed for any option tag. Ex) <sip:<From>>;;;;;=token
8	Overflow-bracket	Only one bracket (< or >) is allowed.
9	Overflow-at	Only one at (@) is allowed.
10	CRLF(Carriage Return/Line Feed)	Every single line should have only one CRLF at the end.

of them are associated with SIP message header. SIP exceptional cases are categorized in Table 3.

To simulate 2426 test cases of PROTOS, we implemented an application, VoIPAttacker, which is capable of sending specific range of PROTOS test cases. Input values are in the range of PROTOS file names, e.g. 000001-000100. Figure 5 (left) shows VoIPDefender detects PROTOS malformed cases from 1 to 193 which are a part of case group number 1; overflow-general, overflow-space, overflow-null, format string, UTF-8 and ansi-escape. `SIP message view` dialog box in Fig. 5 (left) shows detail header field information of 193th test case, which does not have a method name in the first line.

While testing the PROTOS exceptional cases, we found that there are a number of ambiguous cases in the middle of valid forms and invalid forms. For example, `aaaaa@sip.invalid.com` can be a valid URI form, but it is included as an exceptional case in the PROTOS suite. Thus, we identify those 217 cases as legitimate SIP messages, so the total exceptional cases are 2209. When applying original RFC 3261 rules, 1837 of 2209 (74%) exceptional cases are detected as malformed messages while our secure rules detects 100% of them. Figure 5 (right) indicates how many exceptional cases are detected by each rule. The group ID in Fig. 5 (right) is the same as the one in Table 3.

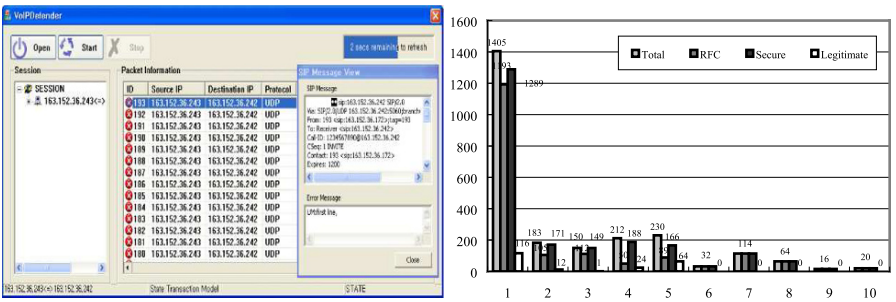

Fig. 5 VoIPDefender (left) and the comparison between original rules and secure ones (right).

5.2 The Result for SIP Flooding Attacks

Before explaining the result, we would like to mention the interesting things that we found while we were testing existing VoIP services. Each VoIP service has been adding specific message header fields such as `PortaBilling` for billing information in Globe7. Vbuzzer is also using `Warning` header fields to transmit noisy feedback. Gizmo Project also defines extra header fields, `JabberID`, `CQBM` and `RemoteIP`. On the other hand, VoIPGo uses a format string when there is a space in a user name. For example, if user name is `voip go`, it is going to change to `voip%20go` because 0x20 is the ASCII code for the space character. Format string is also included PROTOS exceptional cases, so that it may cause erroneous operation.

The most significant fact for SIP flooding detection is how to decide the threshold. The threshold is not supposed to disturb existing VoIP services. Figure 6 (left) depicts the number of transmitted SIP messages for each existing VoIP service.

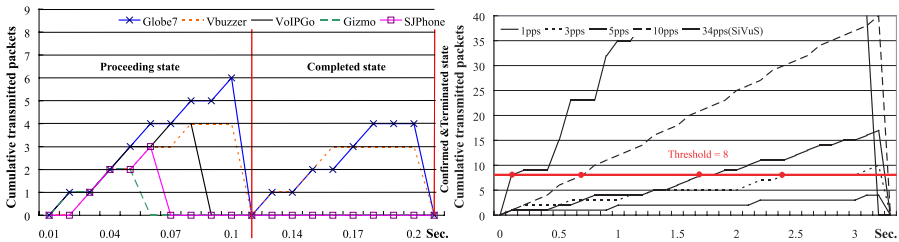

Fig. 6 Number of SIP messages for each state (left) and the result of SIP flooding test (right).

To find an appropriate threshold, we employed the proposed mechanism in the UAC part of SIP system and monitored SIP messages during call-setup process and distinguish the messages according to state. It shows that all five VoIP services send SIP messages under 6 pps (packet per second) per state. From the result, we

could infer how many SIP messages were transmitted under the normal VoIP service condition.

To simulate flooding attack conditions, we applied five different pps (packet per second) cases in SiVuS; 1pps, 3pps, 5pps, 10pps and 34pps. Generating one packet per second is not a big burden in current computer system, but over 3pps starts consuming computer resources.

Figure 6 (right) shows SIP flooding simulation. 1pps is under the threshold, so that it is regarded as a normal condition. Actually it stands to the reason that 1pps is not flooding attack condition because it consumes just little resources. However, 34pps, 10pps, 5pps, and 3pps flooding tests reach to the threshold respectively at 0.2, 0.8, 1.9, and 2.3 second. Using the threshold, we detect flooding attack in 2.3 second that allows only ringing once.

We assume that there is no packet missing and retransmission. Under our experimental environment, small VoIP network between UAC and UAS, proper threshold is 8pps. It means that the average number of transmitted SIP messages from an initial state to its terminate state are normally lower than 6pps. We give 2pps gap as a tolerable range between threshold (8pps) and estimated max value (6pps) because the range is wide enough to reduce false alarm in our assumption. However, there is a possibility to transmit SIP messages more than the threshold under the larger VoIP networks. To adopt different environment, dynamic threshold is necessary but the principle of proposed approach is still useful.

5.3 The Overhead of Proposed Mechanism

We implemented an application, VoIPDefender, based on our detection mechanism. The developing environments are as follows: 3.0 GHz CPU, 2GB DDR2 memory, Windows XP service pack 2, Visual studio 2005 and MFC.

We estimate how many memory it requires and how long it takes to load the rules. VoIPDefender requires about 11 MB to and it is light enough to load for most systems. In fact, 11MB is not necessary because most of 11 MB is used for GUI (Graphic User Interface) such as dialog and window controls. It implies that there is the possibility of reducing the resource consumption. Moreover, it takes only 0.015 second and 352 KB to load the rules and creating session needs 40 KB. As a result, it turns out that VoIPDefender does not consume too much resources, so that it is suitable for applying to modern computer systems.

5.4 The Comparison with The Other Approaches

We presented that proposed mechanism is capable of detection two main SIP attacks in the previous Subsects. 5.1 and 5.2. Furthermore, Table 4 shows our proposed

approach is able to detect additional SIP attacks compared with existing similar approaches. Three additional SIP attacks that proposed approach covers are follows.

- Invalid header field: a message missing the mandatory header or containing the non-allowed header.
- Improper message transmission: a message that is unacceptable to current state.
- Session information mismatch: a message containing wrong `CSeq` or `Call-ID`.

Neither rule matching nor state machine approach detects any of three SIP attacks. Also, simple combination approach of rule matching and state machine only covers two main SIP attacks, malformed and flooding attacks. However, proposed approach covers all SIP attacks by using SIP features, and shows higher detection rate for malformed SIP attack as applying secure rule sets that we developed.

Table 4 The comparison with the other approaches

Attack/Approach	Rule matching	State transition	Simple combination (Rule+State)	Proposed approach
Malformed	74%	X	74%	100%
Flooding	X	O	O	O
Invalid header field	X	X	X	O
Improper message transmission	X	X	X	O
Session information mismatch	X	X	X	O

6 Conclusion

We propose a complementary mechanism for detecting both malformed SIP messages and SIP flooding attacks. Moreover, proposed mechanism covers three additional SIP threats and shows 26% higher detection rate for malformed SIP attacks. To sum up, there are three strengths of proposed mechanism. First, the secure rules that we propose show the improvement apparently for detecting malformed SIP messages than original RFC 3261 ones. Also, the result shows that all PROTOS malformed SIP messages can be detectable by our rule matching algorithm, and it is confirmed that the algorithm is effective to protect VoIP services from variant malformed message attacks. Second, we modify the original state transitions and utilize a threshold based on practical VoIP services. Proposed state transition models with the threshold have not interrupted existing VoIP services, and it is possible to recognize flooding conditions. Lastly, through using SIP features from the rule sets and state machines, proposed mechanism catches three more SIP attacks; invalid header field, improper message transmission, and session information mismatch.

As a consequence, we insist that it is possible to build more robust the VoIP systems by applying our proposed mechanism. Furthermore, our mechanism can be adopted as a lower layer detection module to protect higher layer VoIP applications.

For future works, we have a plan to extend the rule matching algorithm to apply for SDP (Session Description Protocol) because the header fields of SDP are also plain texts. In addition, we will study how to apply the proposed approach to a complicated network system, such as a system with SIP proxy servers and gateways.

References

1. Chen, E.: Detecting DoS attacks on SIP systems. In: Proc. of VoIP Management and Security (2006)
2. Computer Engineering Laboratory, University of Oulu: PROTOS Test-Suite:c07-sip (2005). URL http://www.ee.oulu.fi/research/ouspg/protos/testing/c07/sip/index.html
3. Geneiatakis, D., Kambourakis, G., Dagiuklas, T., Lambrinoudakis, C., Gritzalis, S.: A framework for detecting malformed messages in SIP networks. In: Proc. of Local and Metropolitan Area Networks (LANMAN) (2005)
4. Handley, M., Jacobson, V.: RFC2327: Session description protocol (SDP) (1998)
5. Hung, P., Vargas Martin, M.: Security issues in VoIP applications. In: Proc. of Electrical and Computer Engineering, Canadian Conference (2006)
6. Ilgun, K., Kemmerer, R., Porras, P.: State transition analysis: A rule-based intrusion detection approach. IEEE Trans. on Software Engineering (1995)
7. McGann, S., Sicker, D.: An analysis of security threats and tools in SIP-based VoIP systems. In: Proc. of the 2nd Workshop on Securing Voice over IP, Cyber Security Alliance (2005)
8. MyVoIPProvider.com: Rank and Compare the Worlds Top 100 VoIP Providers (2007). URL http://www.myvoipprovider.com/
9. Packetizer, Inc.: H.323 versus SIP: A comparison (2007). URL http://www.packetizer.com/voip/h323_vs_sip
10. Sengar, H., Wijesekera, D., Wang, H., Jajodia, S.: VoIP intrusion detection through interacting protocol state machines. In: Proc. of Int'l Conf. on Dependable Systems and Networks (DSN) (2006)
11. Sisalem, D., Kuthan, J., Ehlert, S.: Denial of service attacks targeting a SIP VoIP infrastructure: attack scenarios and prevention mechanisms. IEEE Network (2006)
12. Vigna, G., Kemmerer, R.: NetSTAT: A network-based intrusion detection approach. In: Proc. of the 14th Annual Computer Security Application Conference (ACSAC) (1998)
13. Vigna, G., Robertson, W., Kher, V., Kemmerer, R.: A stateful intrusion detection system for world-wide web servers. In: Proc. of the Annual Computer Security Applications Conference (ACSAC) (2003)
14. Voice over Packet Security Forum: SiVuS: the VoIP Vulnerability Scanner (2006). URL http://www.vopsecurity.org/html/downloads.html
15. Walsh, T., Kuhn, D.: Challenges in securing voice over IP. IEEE Security & Privacy (2005)
16. Wu, Y.S., Bagchi, S., Garg, S., Singh, N.: SCIDIVE: a stateful and cross protocol intrusion detection architecture for voice-over-IP environments. In: Proc. of Int'l Conf. on Dependable Systems and Networks (DSN) (2004)