Journal of Advances in Computer Research
Quarterly pISSN: 2345-606x eISSN: 2345-6078
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 7, No. 3, August 2016), Pages: 33-45
www.jacr.iausari.ac.ir

Intrusion Response System for SIP based Applications with Engineered Feature Set

Hassan Asgharian✉[1], Ahmad Akbari[1], Bijan Raahemi[2]

1) Computer Engineering, Iran University of Science and Technology, Tehran, Iran
2) School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Canada

asgharian@iust.ac.ir; akbari@iust.ac.ir; braahemi@ottawa.ca

Received: 2015/12/18; Accepted: 2016/02/24

Abstract

Session Initiation Protocol (SIP) is the main signaling protocol of next generation networks (NGN). SIP based applications are usually deployed over the Internet, for which their text-based nature and internal stateful operation make them vulnerable to different types of attacks. The real-time functionality of SIP based applications make their related security systems more complex. On the other hand, automatic response to intrusions is one of the most important issues in securing different applications. The current state of intrusion detection systems (IDS) is that they often generate too many same or similar alerts for one intrusion which makes the function of response system unreliable. In this paper, we propose a security framework for automatic intrusion response in SIP environments. Our framework consists of specific firewall, detection engine and response part. The SIP firewall works based on URIs (universal reference identifier), and filters the incoming packets in the edge of network. Input packets are directed to the specification based detection engine which works based on the proposed exactly engineered features. The output of this system and the current state of the SIP proxy (e.g. call completion rate, call rejection rate and etc.) are fed to the response system to make a final decision. A prepared test bed is used for analyzing the performance of the proposed response system, measuring its performance using three distinct datasets. The experimental results show the performance of the proposed response system in terms of detection rates.

Keywords: *SIP IDS, flooding attacks, NGN and IMS security, Intrusion Response System*

1. Introduction

Session Initiation Protocol (SIP) is used for creating, modifying and terminating user sessions for multimedia data [1, 2]. SIP entities are vulnerable to flooding attacks, because of both implementation and protocol weaknesses [3, 4]. These threats can be divided into external and internal ones [5]. External threats are typically launched by a non-participant in a SIP-based communication, while internal threats originate from within the SIP network. Furthermore, the text-based nature of SIP messages and SIP's stateful function affords attack opportunities such as message tampering and application layer denial of service (DoS) [6, 4].

A requirement for Intrusion Detection Systems (IDS) is the ability to collect and analyze network traffic, and classify it as either normal or abnormal behavior [5]. When an intrusion is detected, it is desirable to take action to thwart the attack, and ensure the safety of the network environment. Such countermeasures are referred to as an *intrusion*

response. Although an intrusion response component is often integrated with an IDS, it receives considerably less attention in the context of security of next generation networks [7]. However, response generation in real-time environments like SIP networks is more complicated than for other types of networks. Intrusion response systems (IRS) are designed to respond at runtime to the attack in progress and can be classified according to these characteristics [7]:

- Activity of triggered response (passive or active)
- Level of automation (notification, manual response and automatic response)

Because of the importance of automatic response systems, these systems can be further classified [8]:

- Ability to adjust (static or adaptive)
- Time instance of the response (proactive or delayed)
- Cooperation capability (autonomous or cooperative)
- Response selection mechanism (static mapping, dynamic mapping and cost-sensitive)

Although it is possible to consider SIP user agents as the victims of possible attacks, we will refer to the SIP proxy server itself as the target entity of attacks, since SIP proxies are typically the most valuable assets of service providers. In this paper, we propose an active response system which adapts its actions based on environmental conditions, working autonomously by selecting appropriate responses from a set of nominated alternatives. We also extract a minimal feature set for high detection rates and low false alarm rates in our IDS. Since the traditional detection solutions rely on standard misuse- or anomaly-based detection technologies are incompatible with the limitations and monitoring requirements of SIP networks, we introduce a specification-based intrusion detection sensor to accurately monitor the traffic at the edge of an SIP network. Reliance on specifications rather than attack signatures or statistical profiles enables our solution to detect unknown threats while providing detailed information about malicious behaviors detected.

The paper is structured as follows. In section 2, a review of previous work is presented, focusing on SIP response systems. The proposed response framework and our engineered feature set are described in section 3. Evaluation and analysis of experimental results from testing the system are presented in section 4, including an overview of the dataset employed. Finally, we present our conclusion and discuss opportunities for future work in section 5.

2. Related Works

One of the most important issues in intrusion detection is automating responses to intrusions [9]. Intrusion detection analysis approaches are classified to misuse (or signature), anomaly (statistical) and specification [6, 9]. The traditional detection solutions rely on standard misuse- or anomaly-based detection technologies are incompatible with the limitations and monitoring requirements of SIP networks. For

example, the lack of information about existing attacks in SIP environments and the need to be able to detect unknown attacks make it impossible to use signature-based solutions. On the other hand, the need to rapidly understand the root causes of attacks limit the use of anomaly-based solutions [3]. Therefore we introduce a specification-based intrusion detection [10] sensor to accurately monitor the traffic at the edge of an SIP network. Reliance on specifications rather than attack signatures or statistical profiles enables our solution to detect unknown threats while providing detailed information about malicious behaviors detected [11]. Specification based techniques avoid the typically high rate of false alarms caused by legitimate but unseen behavior in the anomaly detection approach. However, development of detailed specifications can be time-consuming and also be protocol and application-specific [6]. The other shortcoming of this technologies is that the specifications are often very difficult to evaluate and verify [11].

Authors in [12] introduced a new approach to detect CPU-based DoS attacks by using the weaknesses of SIP authentication mechanism and cut off the access of intruders. This type of system is located at the edge of a SIP network, automatically collects user profiles, and acts as an independent specification based detection unit equipped with automatic black list based response system. Another access list based prevention system is proposed in [13] that used against flooding attacks. It uses whitelist approach and deploy traditional bloom filter approach in its detection phase. Using bloom filters in detection of flooding based attacks is also proposed in [14]. The main problem with bloom filter approaches is that they have not acceptable performance in complicated attack scenarios like low rate SIP memory attacks. Authors in [15] implemented a SIP intrusion detection and response framework for classifying incoming SIP traffic, thus limiting access by detected intruders to the SIP server. This framework has detection capability, but limited response action. Their proposed response module only writes notification alarms into a file, and blocks the suspicious access to the system. Another specification-based intrusion detection framework based on a SIP finite-state machine to distinguish deviation from normal or expected behavior is proposed in [16]. Through communication with a firewall component, this scheme allows blocking of offending traffic, thus keeping the system in service even under attack conditions.

Each of these approaches has advantages and disadvantages. Some have only detection mechanisms, but no built-in response. Some trigger responses by using firewalls, but none base responses on related operational cost of a response in a given environment and response goodness with respect to detected intrusions and finally the response impact on the system. Those with active responses are triggered to minimize the intrusion damage, while passive response types notify the system administrator and provide attack information. Notification systems generate the alert when an attack is detected, which can contain information about the attack, such as a description, a time of attack, a source IP, user account attributes, etc. Alerts can be used by an administrator or other smart entity (e.g. autonomous IRS) to select an appropriate response. A major challenge to overcome is the delay between intrusion detection and a human response [17]. Manual response systems employ preconfigured sets of responses, each mapped to specific attack types. This approach is more highly automated than the notification system approach [17]. Automatic response systems are designed to be fully automated, so that no human intervention is required, and so no delay between intrusion detection and response. With these design goals and limitations

in mind, the characteristics of the response system that this paper proposes are highlighted in Figure 1. This taxonomy is introduced in [7].

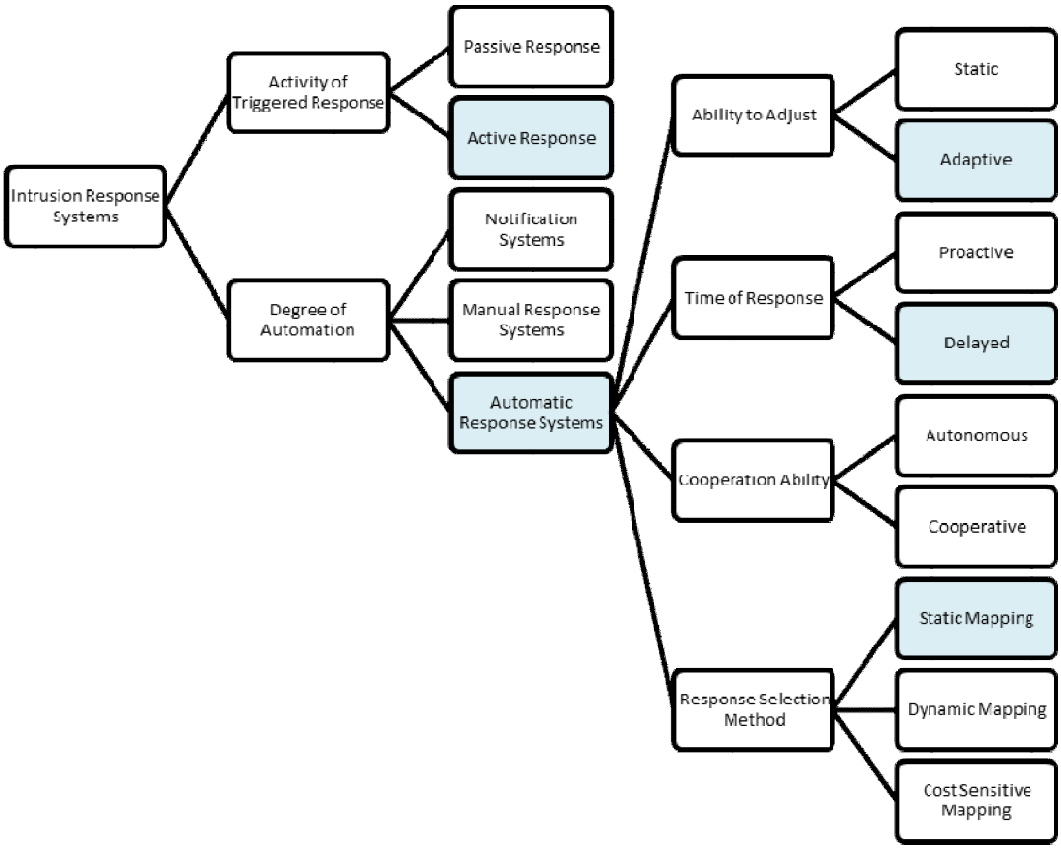

Figure 1. Chosen characteristics of the proposed response system, based on the IRS taxonomy [7]

We proposed a SIP response system that extracts specifications as a specific feature set based on known SIP vulnerabilities. These specifications are used in our detection system to separate SIP flooding attacks. Decisions to apply a response are made via the SIP firewall, based on the current condition of the system. Details on this process are described in the following sections.

3. Automated Intrusion Response System

A simplified SIP call process known as a basic SIP trapezoid [5] is shown in Figure 2. All SIP calls are started with an initial request (e.g. INVITE) followed by corresponding responses (e.g. 1xx) to form a media session. This process terminated by another request (e.g. BYE) and its related responses.

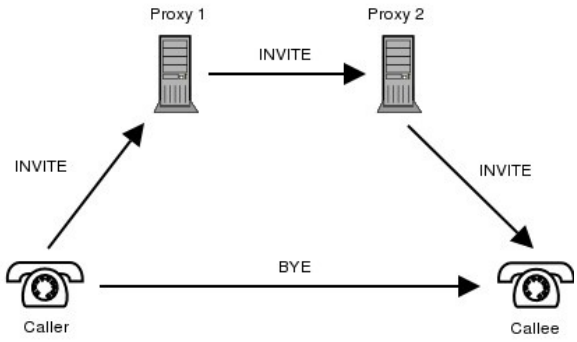

Figure 2. A basic SIP trapezoid

Since all SIP entities have to accept and analyze the incoming INVITE requests (risky or otherwise) [1], an intruder can take advantage of this in arranging an attack on the SIP entities based on this request. Therefore, in real world applications finding a way to make SIP connections secure is important. Because of the complexity, real-time nature and stateful functionality of SIP, development and deployment of security components in SIP environments has some challenges. Strategic placement of the response, monitoring and other security elements of the system are crucial requirements. There are multiple methods available to connect an IDS to monitor and analyze traffic (e.g. using a hub, a network tab, or inline deployment), but in practice a response system should always connected inline. This requirement enables the IRS to drop selected packets, and to defend against an attack before it can assume control of the internal network. Our proposed response architecture is shown in Figure 3. It is placed in entrance point or edge of the network.

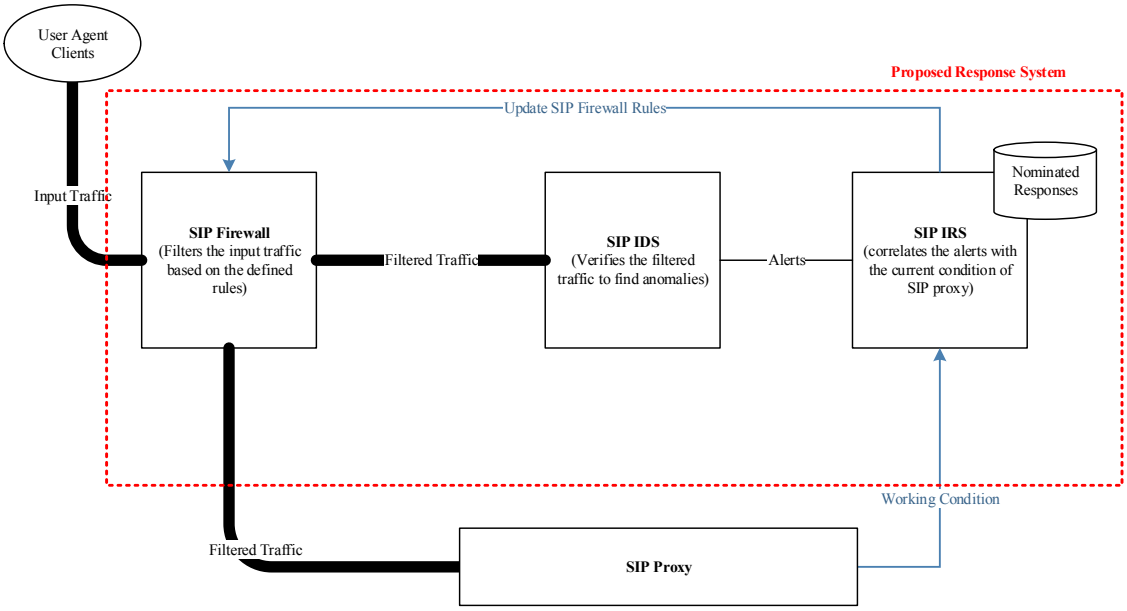

Figure 3. Conceptual architecture of the proposed response system

The presented SIP response system has three main general security components specialized for SIP: firewall, IDS and IRS. We also include a management interface to update the firewall rules for response automation. Therefore the system self-adapts to

defend against the latest threats (see Figure 3). A data state diagram of the proposed architecture is shown in Figure 4. Input SIP packets are passed through our inline specific application layer firewall, to be filtered based on the SIP header fields (e.g. SIP URI). The output of the SIP firewall is directed to the SIP intrusion detection module for analysis and traffic verification. The intrusion detection system generates alerts based on the SIP specification and the response module integrates the alerts with conditioning variables of the SIP server to make filtering rules. The generated rules are fed to the application layer firewall to filter further incoming traffic.

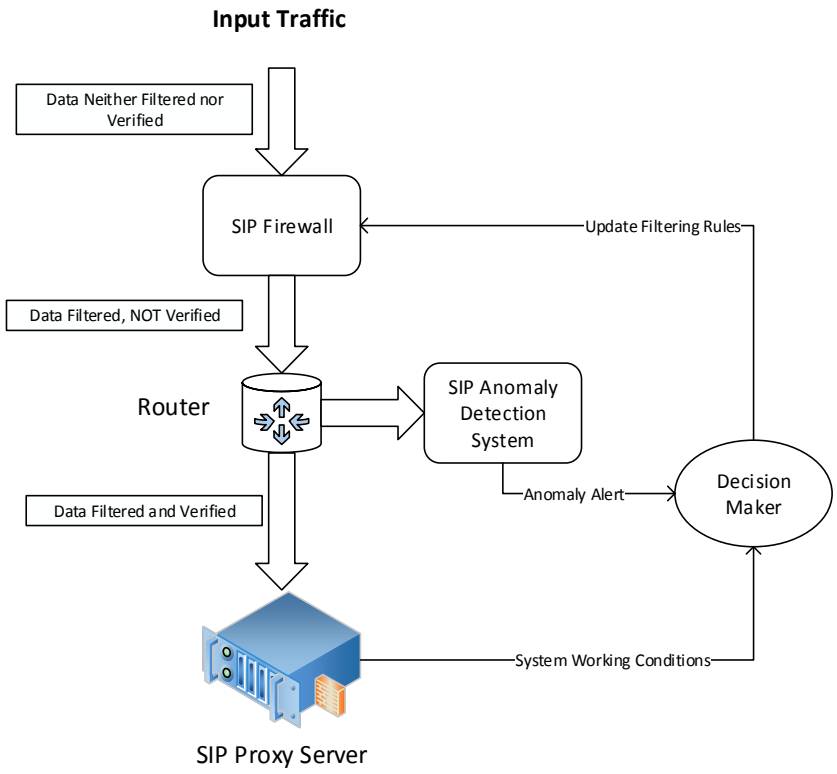

Figure 4. Data state diagram of the system

3.1 Elements of the Proposed Response System

The main components of the proposed architecture are shown in Figure 5:

- The SIP firewall is located inline with proxy server to monitor and filter the input traffic. Its filtering rules are updated automatically by the IRS via a configuration interface.
- The IDS deploys the optimized engineered feature set as specification with a novel detection engine. It detects any deviation from normal behavior, and sends appropriate alarms to the IRS after receiving filtered traffic from the firewall. The detection engine uses the normal model of the system.

- The IRS receives its inputs from the IDS, and correlates its alarms with environmental parameters of the SIP proxy server. These parameters include the call completion rate, the call setup time, and the response time of the proxy server. The decision maker module selects one of the following nominated responses, based on the working conditions:
    - Simple and conservative: sending notification only.
    - Complex active response: limiting the access of some users based on their SIP URIs.

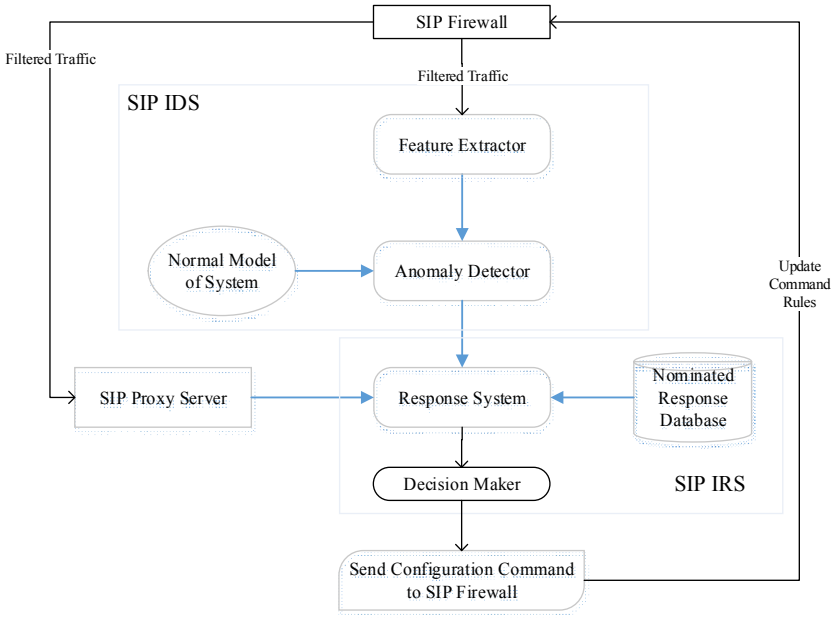

Figure 5. Main components of the response framework

In the following subsections, more details about the security components are presented.

3.1.1 SIP Firewall

The SIP firewall works in the application layer, filtering input SIP packets based on its access list (whitelist or blacklist). This offers one (or more) of the following options in a specific time window:

- The destination SIP_URI ('To' tag)
- The source SIP_URI ('From' tag)
- The transaction identifier (VIA Branch and CSeq)
- The source initiation network (IP address)

As shown in ***Figure 4***, the input traffic and filtering rules are fed to the SIP firewall. The SIP firewall accepts the incoming commands to block specific users and networks. It also accepts the transaction identifiers, so they can be closed. The firewall rules are set automatically by the IRS in our proposed architecture. All registered firewall rules have a specific expiration time and they remove automatically from firewall rule set. If

the expiration times of these rules are met, they will be cleared from the rule database. The expiration times are set based on the administrator's policies, and may be related to previous user activity. This simplifies and automates administration using the proposed architecture.

3.1.2 SIP Intrusion Detection System

In our previous work [6], we proposed a feature set to use in a SIP IDS. Since the presentation of an active response method with regard to environmental situations is the main goal of the current work, it is necessary to have minimum detection time with no false alarms. Accordingly, we select a minimum set of features with maximum coverage on determined anomalies (specifically, flooding attacks) and arrange a specification based detection engine based on the engineered features. We only considered the average and variance of each proposed feature under normal conditions as a specification of the system without having any precondition about the input call rates. In other words, our proposed features model the normal behavior of SIP entities. The engineered features to detect SIP flooding attacks with short description about their attack target are shown in *Table 1.*

Table 1. Proposed Engineered Features

#	Feature	Target
1	$\frac{Number\ of\ \mathrm{Re}quests}{Number\ of\ Packets}$	Basic and Advance Flooding Attack
2	$\frac{Number\ of\ Requests}{Number\ of\ Responses}$	Basic and Advance Flooding Attack
3	$\frac{Number\ of\ Responses}{Number\ of\ Packets}$	Basic and Advance Flooding Attack
4	$\frac{Number\ of\ INVITEs}{Number\ of\ Requests}$	Invite Flooding Attack
5	$\frac{Number\ of\ 4xx}{Number\ of\ Responses}$	Register and other Response Flooding
6	$\frac{Number\ of\ 2xx}{Number\ of\ Requests}$	Advance Flooding Attack
7	$\frac{Number\ of\ Transactions}{Number\ of\ Packets}$	Advance Flooding Attack
8	$\frac{Number\ of\ Branches}{Number\ of\ Packets}$	Advance Flooding Attack
9	$\frac{Number\ of\ Recievers}{Number\ of\ Requests}$	Basic and Advance Flooding Attack

During run time, we compare the value of the corresponding specification (i.e. feature) with its average in normal conditions. If this rate is less than the variance in normal situations, the traffic is tagged as normal and otherwise an alert will be generated. The proposed detection engine is constructed by using one-class classification, also known as unary classification or novelty detection, which tries to identify objects of a specific class amongst all objects. A block diagram of the proposed

detection system is shown in *Figure 6*. A time-based buffer was used to collect input SIP messages. The window timeout was tuned based on input traffic rate and the system processing power. Since detection is done within each window, the maximum time delay of the system is equal to the window size. A larger timeout value caused the response system react with increased delay.

In the learning phase, a training set containing only the normal instances should be used. This is more challenging than the traditional classification problem, which tries to discriminate two or more classes for a training set containing objects from all classes. We use a simple detection technique by using the engineered features to verify the input traffic and compute the mean of each feature under normal conditions as a baseline reference, and compare its difference with the runtime value. If this value exceeds the variance of the feature in a normal state, an alert is generated. With simultaneous use of the proposed features, a simple voting mechanism is deployed to make decisions about the nature of the traffic (benign or malicious).

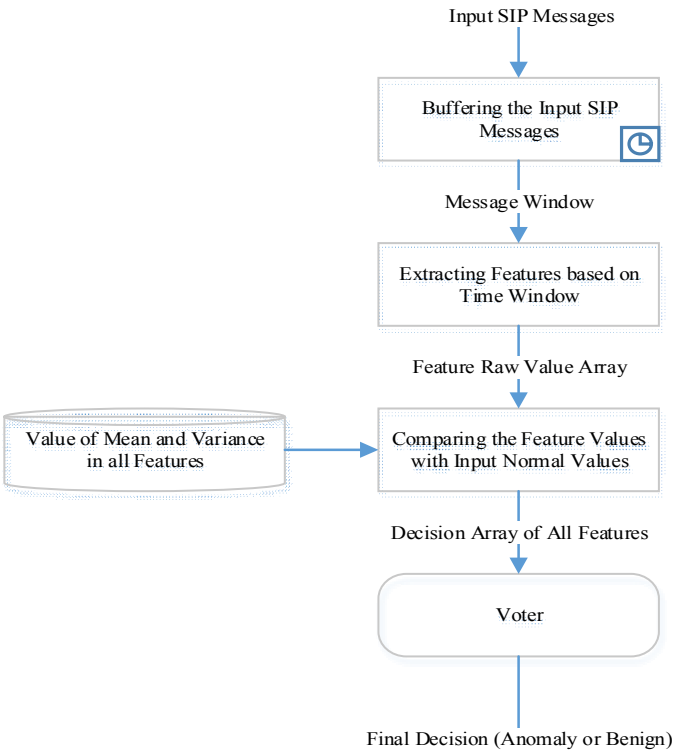

Figure 6. The procedure of proposed detection engine

3.1.3 SIP Intrusion Response System

The proposed SIP IRS has two input interfaces, one with the SIP proxy server and the other with the SIP IDS and it has one output interface to SIP firewall. Incoming calls are forwarded to this module from the SIP IDS. The output of the proposed IDS is the decision of the detection engine about the current status of input requests (benign or attack). Since the response is to be adapted based on the current status of the SIP proxy server, the following metrics are considered in the decision procedure regarding the type of response:

- A performance metric of the proxy server (e.g. call setup time, call rejection rate, call response time, CPU and RAM use)
- Possible side effects of the response (such as terminating valid SIP sessions)

The system can accommodate detection engine faults related to false negatives and false positives, which makes the process of the detection engine simpler. In other words, if the state of the proxy server in terms of resource utilization and other performance metrics is suitable, then there is no need for applying a response. Accordingly, the current usage of main resources (CPU, RAM and bandwidth) and also the current completion and rejection rates as well as call setup times of the SIP proxy are considered. If these rates and times are acceptable, there is no need to react against the detected attacks because the system is still working normally. Considering the parameters described, the response alternatives are as summarized in Table 2:

Table 2. Response alternatives

#	Response action	Positive effects	Side effects
1	Notification	Informs the administrator about the risks of the current condition	None
2	Cut access for a specific offending user	Prevents access of users to the system by making a blacklist of SIP URIs	Blocks the authorized user
3	Cut access for a specific annoying network	Prevents access of a group of users to the system by making a blacklist of source IP addresses	Blocks the authorized network and users
4	Close unused and abnormal open transactions	Decreases the resource utilization values	None

4. Experimental Setup and Result Analysis

Since the most frequent SIP attack type is flooding [18], we arranged our experimental setup based on available flooding datasets. The first dataset was collected using our test-bed in NRG, containing different flooding attacks (NRG-IUST) [19, 15]. Implementation details of the traffic generation were described in our previous published work [19], [15] (which are available online). This dataset contains all known SIP flooding attacks, including INVITE, REGISTER, RINGING, as well as mixtures of them. We also used two other datasets were taken from INRIA, based on two different SIP proxy servers, OPENSIPS and ASTERISK [20]. These datasets also contain INVITE-based flooding attacks. The datasets are summarized in Table 3.

Table 3. Summary of datasets

#	Title	Number of Packets	Total Time of Dataset (sec)	Description
1	NRG-NA	18702	300	This dataset contains a mixture of SIP flooding attacks.
2	NRG-INVITE	20243	300	This dataset contains an INVITE SIP flooding attack in advance and basic scenarios.
3	NRG-RINGING	16287	300	This dataset contains a RINGING based SIP attacks which deplete the victim's memory.
4	INRIA-OPENSIPS	23396	180	This dataset contains an INVITE SIP flooding attack in the basic scenario on OPENSIPS.
5	INRIA-ASTERISK	2109	180	This dataset contains an INVITE SIP flooding attack in the basic scenario on ASTERISK.

Experiments were designed to assess the system capabilities in terms of intrusion detection after applying responses. Tests were performed with three different window sizes on all input traffic to show the effectiveness of the proposed architecture. The shown detection rates of the different datasets in *Figure 7*, were selected from receiver operating characteristic (ROC) with no false alarms. This helps the response system to have better performance which is shown the capability of proposed minimal feature set.

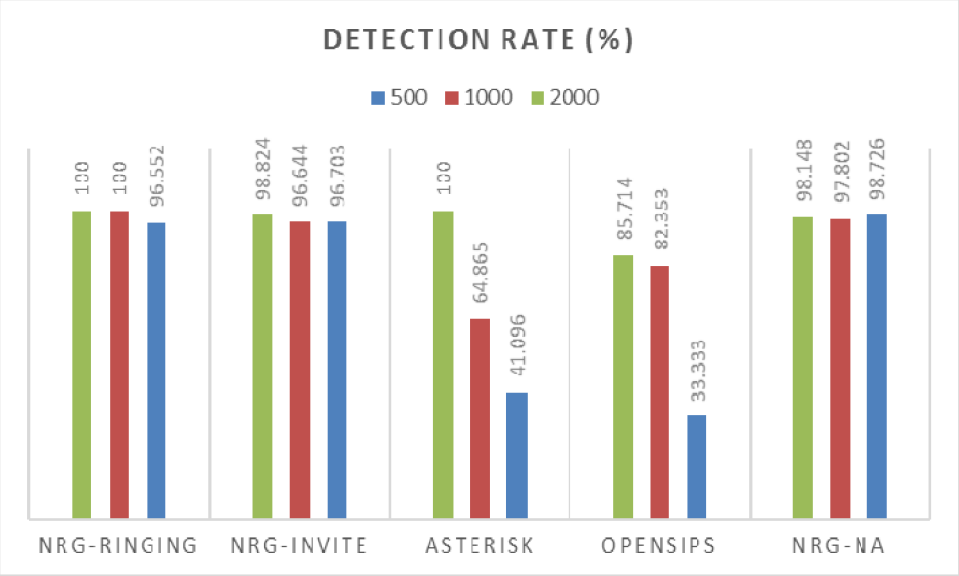

Figure 7. Detection rate of the proposed detection system with no false alarms

The performance of the IRS is tested by using its feedback to the SIP firewall to add appropriate rules to filter the input traffic before being verified in intrusion detectors. Figure 8 shows the output of the system when the SIP firewall was used to filter input traffic.

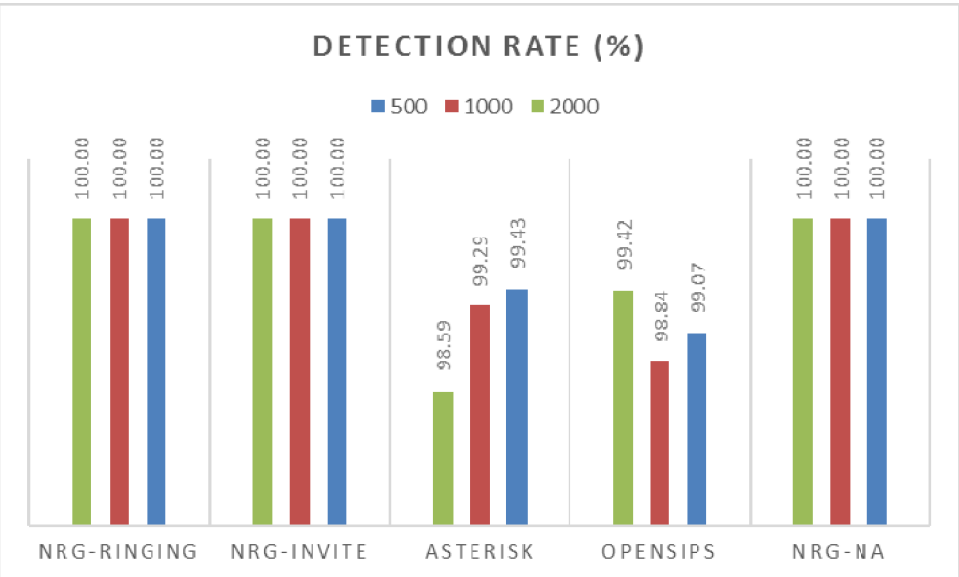

Figure 8. Detection rate after activating response feedback

The experimental results show that it was possible to filter almost all unwanted input traffic by appropriate deployment of the proposed architecture in a SIP-based system in real time. The proposed architecture minimizes the false alarm rate, and protects the SIP entity through selection and processing of runtime parameters, as well as real time filtering of input traffic.

5. Conclusion

We propose an automated real time IRS to make SIP proxy servers more secure. Our novel architecture consists of simultaneous use of three SIP security components: a firewall, an IDS, and an IRS. The firewall filters input SIP packets based on their SIP attributes (e.g. transaction attributes, and sender and receiver information). This component is installed inline, and works at the packet level. Firewall rules are updated automatically by the IRS. The IRS receives alerts from SIP IDS and correlates these alerts with the server working conditions to make firewall filtering rules. The IDS verifies the filtered traffic to separate benign from attack packets by deploying a novel proposed feature set. The engineered features are fed to our intrusion response framework, which is a combination of a specification-based SIP IDS, an application layer SIP firewall, and a SIP-specific response selection engine. We plan to extend our research on the SIP response systems, considering the response related costs through different decisions among different elements of the framework.

References

[1] *RFC3261, SIP: Session Initiation Protocol,* 2002.

[2] S. A. Baset, V. K. Gurbani, A. B. Johnston, H. Kaplan, B. Rosen and J. D. Rosenberg, "The Session Initiation Protocol (SIP): An Evolutionary Study," *JOURNAL OF COMMUNICATIONS,* vol. 7, pp. 89-105, 2012.

[3] A. D. Keromytis, Voice over IP Security: A Comprehensive Survey of Vulnerabilities and Academic Research, Springer, 2011.

[4] S. Ehlert, D. Geneiatakis and T. Magedanz, "Survey of network security systems to counter SIP-based denial of service attacks," *computers & security,* vol. 29, no. 2, p. 225–243, 2010.

[5] D. Sisalem, J. Floroiu, J. Kuthan, U. Abend and H. Schulzrinne, SIP Security, Wiley, 2009.

[6] H. Asgharian, A. Akbari and B. Raahemi, "Feature Engineering for detection of Denial of Service attacks in session initiation protocol," *Wiley Security and Communication Networks,* vol. 8, pp. 1587-1601, 2015.

[7] N. Stakhanova, S. Basu and J. and Wong, "A taxonomy of intrusion response systems," *International Journal of Information and Computer Security,* vol. 1, no. 1, pp. 169-184, 2007.

[8] B. Foo, M. W. Glause, G. M. Howard, Y.-S. Wu, S. Bagchi and E. H. Spafford, "Intrusion Response Systems: A Survey," in *Morgan Kaufmann Publishers*, 2008 , pp. pp. 377-412.

[9] I. Balepin, S. Maltsev, J. Rowe and K. Levitt, "Using Specification-Based Intrusion Detection for Automated Response," *Lecture Notes in Computer Science; Recent Advances in Intrusion Detection,* pp. 136-154, 2003.

[10] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang and S. Zhou, "Specification based Anomaly Detection: A New Approach for Detecting Network Intrusions," in *CCS*, 2002.

[11] R. Berthier and W. Sanders, "Specification-Based Intrusion Detection for Advanced Metering Infrastructures," in *IEEE 17th Pacific Rim International Symposium on Dependable Computing (PRDC)*, 2011.

[12] S. Pourmohseni, H. Asgharian and a. A. Akbari, "Detecting authentication misuse attacks against SIP entities," in *10th International ISC Conference on Information Security and Cryptology (ISCISC)*, 2013.

[13] B.-h. Roha, J. W. Kimb, K.-Y. Ryub and Jea-Tek Ryuc, "A whitelist-based countermeasure scheme using a Bloom filter against SIP flooding attacks," *Elsevier Computers & Security*, vol. 37, pp. 46-61, 2013.

[14] D. Geneiatakis, N. Vrakas and C. Lambrinoudakis, "Utilizing bloom filters for detecting flooding attacks against SIP based services," *Computers & Security*, vol. 28, no. 7, p. 578–591, 2009.

[15] Z. Asgharian, H. Asgharian, A. Akbari and B. Raahemi, "Detecting Denial of Service Message Flooding Attacks in SIP based Services," *Amirkabir Journal of Technology*, vol. 44, no. 1, pp. 74-81, 2012.

[16] S. Ehlert, C. Wang, T. Magedanz and D. Sisalem, "Specification-based Denial-of-Service Detection for SIP Voice-over-IP Networks," in *3rd International Conference on Internet Monitoring and Protection*, 2008.

[17] A. Shameli-Sendi, N. Ezzati-jivan, M. Jabbarifar and M. Dagenais, "Intrusion Response Systems: Survey and Taxonomy," *International Journal of Computer Science and Network Security*, vol. 12, no. 1, pp. 1-14, 2012.

[18] D. Seoa, H. Leea and E. Nuwereb, "SIPAD: SIP–VoIP Anomaly Detection using a Stateful Rule Tree," *Elsevier, Computer Communications*, vol. 36, no. 5, p. 562–574, 2013.

[19] Z. Asgharian, H. Asgharian, A. Akbari and B. Raahemi, "Detecting Denial of Service Attacks on SIP Based Services and Proposing Solutions," in *Privacy, Intrusion Detection and Response: Technologies for Protecting Networks*, P. Kabiri, Ed., IGI Global, 2012, pp. 145-167.

[20] M. Nassar, R. State and O. Festor, "Labeled VoIP Data-Set for Intrusion Detection Evaluation," *Lecture Notes in Computer Science Networked Services and Applications - Engineering, Control and Management* , vol. 6164, pp. 97-106, 2010.