Battling against DDoS in SIP. Is machine learning-based detection an effective weapon?

Z. Tsiatsikas[1], A. Fakis[1], D. Papamartzivanos[1], D. Geneiatakis[2], G. Kambourakis[1] and C. Kolias[3]
[1]*Dept. of Inform. and Comm. Systems Engineering, University of the Aegean, Karlovassi, Greece*
[2]*Electrical and Computer Engineering Department, Aristotle University of Thessaloniki, GR541 24 Thessaloniki, Greece*
[3]*Computer Science Department, George Mason University, VA, USA*
{*tzisis, alfa, dpapamartz, gkamb*}*@aegean.gr*
dgeneiat@auth.gr
kkolias@gmu.edu

Keywords: Session Initiation Protocol, Machine Learning, DDoS, Anomaly-Detection, Intrusion Detection Systems.

Abstract: This paper focuses on network anomaly-detection and especially the effectiveness of Machine Learning (ML) techniques in detecting Denial of Service (DoS) in SIP-based VoIP ecosystems. It is true that until now several works in the literature have been devoted to this topic, but only a small fraction of them have done so in an elaborate way. Even more, none of them takes into account high and low-rate Distributed DoS (DDoS) when assessing the efficacy of such techniques in SIP intrusion detection. To provide a more complete estimation of this potential, we conduct extensive experimentations involving 5 different classifiers and a plethora of realistically simulated attack scenarios representing a variety of (D)DoS incidents. Moreover, for DDoS ones, we compare our results with those produced by two other anomaly-based detection methods, namely Entropy and Hellinger Distance. Our results show that ML-powered detection scores a promising false alarm rate in the general case, and seems to outperform similar methods when it comes to DDoS.

1 Introduction

During the last years Voice over IP (VoIP) technologies and services have penetrated the market and for many of us became an integral part of our software and/or hardware portfolio. Recent reports indicate that this market will grow to reach about USD 136.76 billion until 2020 (Mohr, 2014). In both mobile and fixed networks, Session Initiation Protocol (SIP) seems to be the predominant means for establishing and managing a VoIP session. On the downside, the text and open nature of the protocol has given rise to a plethora of attacks against it.

By examining the rather rich literature on SIP security, one can distinguish several categories of assaults ranging from SQL injection to Denial of Service (DoS) (Geneiatakis et al., 2005; Geneiatakis et al., 2007; Geneiatakis et al., 2006; Kambourakis et al., 2011). It can be safely argued that the latter category attracts the greater attention, and seems to be the most perilous and difficult to confront since it is closely related with the signaling nature of the protocol per se. So, focusing on this kind of attacks, so far, several protection and detection methods have been proposed. Roughly, we can categorize them into misuse-detection and anomaly-detection ones. Generally, the first family of methods monitors network activity with exact signatures of known malicious behavior (e.g., observe the network traffic for singular byte sequences), while the second possess a knowledge of normal activity and warns against any deviation from that profile. The latter category of methods, which is the focus of this paper, is usually realized by means of tools borrowed from the Machine Learning (ML) community. This refers to algorithms that are first get trained in an either supervised or unsupervised manner with reference input to learn its particulars, and then are fed with unknown input for accomplishing the real detection process. Specifically for SIP, although the DoS threat has been stressed out and dealt by a significant number of researches (Ehlert et al., 2010; Keromytis, 2011), the applicability and effectiveness of ML techniques to cope against such incidents is still being assessed and certainly in need for further development.

Naturally, this is mainly due to the increased overhead that these methods may bear - especially when it comes to real-time detection and a training phase

is required - in comparison to misuse-based or purely statistical ones. Nevertheless, in this work we argue that ML techniques can be particularly fruitful for examining the high-volume log files of a given VoIP realm in an offline fashion if they contain DoS incidents. Also, this category of methods may show better results when used for the detection of low-rate DoS (also known with the term "low and slow"), which although is not used to paralyze the target system at a fast pace, it consumes valuable network, CPU and memory resources. Ultimately, this results to service delays which in turn cause customer dissatisfaction with direct negative results to the provider's market share.

Taking the above into consideration, the focus of this work is on the applicability of ML techniques to track down DoS incidents, paying special attention to DDoS and low-rate ones. The main contributions of this work can be summarized as follows:

- We assess the effectiveness of several well-known classifiers to detect (D)DoS incidents in SIP in terms of false alarms.
- We offer a method to calculate SIP message headers occurrences from a given log file in a privacy-preserving way based on a predefined message window. The output of this process are fed to the ML algorithm as the case may be.
- Our experiments consider both DoS and DDoS attacks materialized in 15 different realistically simulated SIP traffic scenarios, having different characteristics in terms of number of users and calls per second.
- For DDoS scenarios, we provide a comparison between two other anomaly-based detection methods proposed in the literature and ML-powered detection in terms of effectiveness.

The rest of the paper is organized as follows. Section 2 provides an overview of SIP architecture and briefly describes the threat model. Section 3 details on the creation of the classification features used by ML classifiers, while Section 4 elaborates on the experimental results. The related work is discussed in Section 5. Section 6 draws a conclusion and provides some pointers to future work.

2 Preliminaries

2.1 SIP Architecture & Threat Model

This Section succinctly describes the basic parts of an SIP architecture. This is required to familiarize

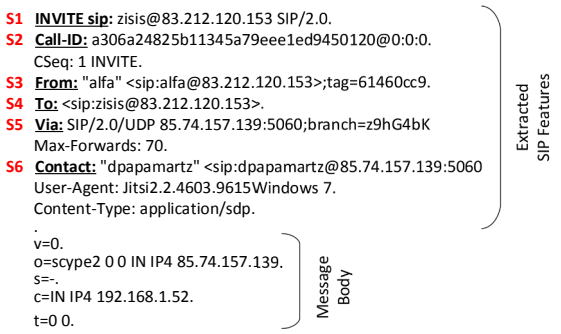

Figure 1: A typical SIP mssage

the reader with the terminology and notations used in the subsequent Sections. An SIP VoIP architecture consists of the following basic elements.

- **User Agent (UA)**: It represents the end points of the SIP protocol, that is, the caller and the callee which are able to initiate or terminate a session using an SIP software or hardware client.
- **SIP Proxy Server**: It is an intermediate entity which plays the role of the client and the server at the same time. Its task is to route all the packets being send and received by the users participating in an SIP session. Note that two or more SIP proxies may exist between any two UAs.
- **Registrar**: It handles the authentication and register requests initiated by the UAs. To do so, this entity stores user's credentials and UA location information.

Figure 1 presents a typical SIP INVITE message. As observed, such a message is consisted of several headers fields, designated as S1-S6 in the figure, and a message body. Initially, a user has to send a REGISTER request in an SIP Registrar. The latter, will store the contact information of the user in the location server. After that, any other user can try to establish a VoIP session with that UA by sending it an INVITE request. At any time, either the caller or the callee can send a BYE message toward the other end to terminate the session. The interested reader who wishes to get a deeper understanding of SIP architecture can refer to the corresponding RFC (Rosenberg et al., 2002).

The SIP signaling produced by the users is logged by the VoIP provider. In fact, this is in most cases a mandatory requirement for any service provider mainly for billing, accounting and network planning purposes. As a result, these logs may be a valuable and rich source of information concerning the investigation of security incidents and intrusion detection in general.

Regarding the security aspects of SIP ecosystems, various types of vulnerabilities and attacks have been presented in the literature so far. More precisely, attacks such as malformed messages, flooding, SQL injection and signaling ones (Geneiatakis et al., 2006; Keromytis, 2012) are some of the most destructive. Among them, (D)DoS is probably the most hazardous one as it targets to drain the target's resources. For example, an attacker is able to send a high volume of requests to the victim with the aim to steer it to paralysis. Moreover, the attacker could send a large number of different requests with spoofed IP addresses, aiming to drain the target's resources and confuse the underlying security mechanisms. In a worst-case scenario, a botnet could be used to launch such an attack, producing high volume of traffic. This may be also orchestrated under the protection of a covert communication channel, thus making the detection even more cumbersome. For a more explanatory threat model on this type of attacks in SIP the reader can refer to (Tsiatsikas et al., 2015).

3 Classification Features

As already mentioned in Section 1, to avoid DoS attacks in SIP several solutions have been proposed (Ehlert et al., 2010; Geneiatakis et al., 2009; Tang et al., 2014). Given that this type of attack is as a rule of thumb executed in a distributed manner and may be quite sophisticated regarding its implementation, simple anomaly-detection approaches that rely on the sudden and fast-paced increment of SIP traffic may be not enough. In this regard, ML-powered methods can be a potent ally towards the detection of such perilous events. The key factor here is the log files on the provider side, which can be used to feed a ML classifier in real-time or offline (in case, say, the investigation of an attack aftermath is required). This Section elaborates on the use of such techniques in an SIP environment.

In our experiments, we utilize and evaluate the effectiveness of 5 well-known classifiers tested under 15 different attack scenarios. Specifically, we use the SMO, Naive Bayes, Neural Networks, Decision Trees (J48) and Random Forest classifiers. This selection has been made based on the ability of these classifiers to perform better in terms of decision accuracy and speed when it comes to numerical data (Witten and Frank, 2005).

In order to take advantage of the aforementioned performance characteristics, we utilize algorithm 1. Its purpose is twofold. On the one hand, it aims to deal with the sensitive nature of the communication transactions residing in an audit trail by providing an anonymization scheme (Tsiatsikas et al., 2015), while on the other allows for automatically extracting the classification features to be used by the classifiers into a numerical form.

The anonymization goal is met using HMAC (Eastlake and Hansen, 2011). HMAC enables one to preserve the anonymity of the communication entities appearing in the underlying audit trail, while the entropy of messages is preserved leading the subsequent calculations to remain intact. In fact, revealing the hidden UA identities is as hard as reversing the HMAC procedure itself. The cryptographic key is kept secret and in possession of the entity, who is the legitimate owner of the audit trail. According to the transformation procedure, a log file is examined line-by-line and every privacy-sensitive SIP message header (e.g., <FROM>, <TO>, <VIA>, etc) becomes input for the HMAC function (lines 2-4). The algorithm considers only the SIP message headers S1 to S6 as given in Figure 1. More precisely, the hash function used in our case is the HMAC-SHA256 one combined with a cryptographic key of 256 bits (line 4).

The next stage is to generate the classification features. The steps to achieve this are summarized in lines 5-14 of algorithm 1. The anonymized unique headers are kept in a Hash table data structure (line 5). This table is populated with the number of occurrences of every single header checksum. That is, if a checksum occurs for the first time, then a new instance is generated in the table (lines 8-9). If it is a repeating header, its number of occurrences is increased by 1 (line 6). This procedure is repeated until a certain message window M_w is met (line 11). In our case, the M_w is set to 1,000, but this parameter can be adjusted by the service provider itself, say, according to the average call rates. To our knowledge, there is no foolproof approach to formally define this parameter, mainly because it is eminently contextual. That is, it is closely connected to the characteristics of the service and underlying network. As a result, similar to other anomaly-based approaches, one can follow an error-trial approach to equilibrate between the M_w parameter and the false alarm rate.

The result of applying algorithm 1 to an audit trail is a number of specially formatted .arff files (one per M_w), which are afterwards used in the classification process. Each .arff file contains classification vectors, i.e., one vector per SIP message found in the log file being examined. Two instances of such a classification vector follows.

$$V_{attack} = \{926, 4, 988, 4, 4, 3, attack\}$$
$$V_{normal} = \{12, 4, 6, 4, 3, 8, normal\}$$

The first 6 values of each vector represent the occurrences of S1 to S6 SIP headers respectively, and the last one characterizes the class in which the vector belongs. One can easily observe that the first vector introduces a higher number of occurrences in S1 and S3 headers, while the rest remain low, close to those contained in V_{normal}.

Algorithm 1: Obtain Input Data for ML Classifiers

```
   Input: Audit Trail
   Output: Input File for Classifiers (.arff format)
1  while (AuditTrail ≠ NULL) do
2      Line ← ReadLine();
3      SIPHeader ← ExtractSipHeader(Line);
4      HashedHeader ← HMAC(SIPHeader);
5      if (InsertToHashTable(HashedHeader) ≠ NULL) then
6          GetValueofHashTable(HashedHeader)++;
7      else
8          InsertToHashTable(HashedHeader);
9          SetValueInHashTable(HashedHeader) ← 1;
10     end
11     if (Message-Window = 1,000) then
12         TotalMessages ← TotalMessages + M_w;
13         Re-Initialize(HashTable);
14     end
15     for (i=1; i ≤ TotalMessages; i++) do
16         PrintOccurences(GetValueofHashTable(HashedHeader));
17     end
18 end
```

4 Evaluation

4.1 Test-bed Setup

In order to evaluate the effectiveness of the aforementioned classifiers in detecting DoS incidents we created a test-bed, depicted in Figure 2. Three different Virtual Machines (VMs) have been used for the SIP proxy, the legitimate users, and the generation of the attack traffic depending on the scenario. All VMs run on an i7 processor 2.2 GHz machine having 6 GB of RAM. For the SIP proxy we employed the widely-known VoIP server Kamailio (Kamailio, 2014). We simulated distinct patterns for both legitimate and DoS attack traffic using *sipp v.3.2*[1] and *sipsak*[2] tools respectively. Furthermore, for the simulation of DDoS attack, the SIPp-DD tool has been used (Stanek and Kencl, 2011). The well-known Weka tool (Hall et al., 2009) has been employed for ML analysis.

As already pointed out in Section 2, we assessed 5 classifiers under 15 different scenarios the results of which is provided in Table 2. It is stressed that both the training and testing scenarios include legitimate and attack traffic. For example, the training scenario is SN1 and its testing scenarios are SN1.1, SN1.2, SN1.3, and so on. The legitimate traffic for DoS testing scenarios was created using the same call rate as that of the corresponding training scenario. On the other hand, for DDoS we used a range of different call rates aiming to better simulate the possible variations that may appear in a real VoIP service. For example, as observed in Table 1, the call rate for SN6.1 is given as 20-120, where the first number indicates the call rate of the attack, and the second corresponds to the call rate of the legitimate traffic both occurring in parallel. Keep in mind that for DDoS scenarios about half of the registered users were generating the normal traffic, while the other half were launching the actual attack. Moreover, for all the scenarios, we employed an exponential inter-arrival time distribution ($\lambda = 100$), for producing the legitimate traffic similar to that used in evaluating SIP server performance (Krishnamurthy and Rouskas, 2013). The attack traffic for DoS training scenarios was created using randomly generated attacks with call rates varied between 20 to 10,000 calls/sec and time pauses between them spanning from 15 to 360 secs. The same method was used for creating the DDoS training scenarios that is, seven variants were launched in total, having different call rates spanning between 2,000 to 15,000 calls/sec and pauses between them set to 10 to 800 secs.

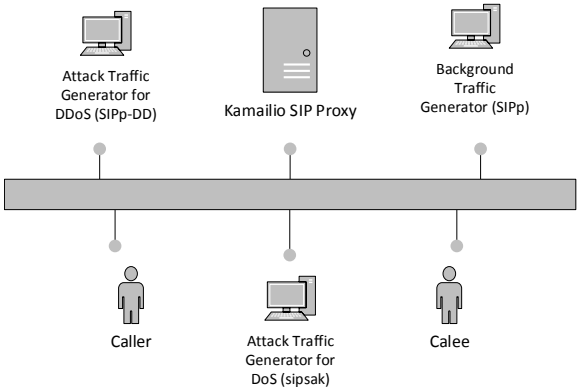

Figure 2: Deployed test-bed for (D)DoS simulations

[1] http://sipp.sourceforge.net/

[2] http://sipsak.org/

Table 1: Description of scenarios

Scen.	Num.of Users	Calls/Sec.	Train Scen.	Type of Attack
SN1	30	2	✓	-
SN1.1	30	50	-	DoS
SN1.2	30	175	-	DoS
SN1.3	30	350	-	DoS
SN2	30	5	✓	-
SN2.1	30	20	-	DoS
SN2.2	30	40	-	DoS
SN2.3	30	80	-	DoS
SN3	30	20	✓	-
SN3.1	30	266	-	DoS
SN4	30	120	✓	-
SN4.1	30	800	-	DoS
SN5	50	120	✓	-
SN5.1	50	400	-	DoS
SN5.2	50	1200	-	DoS
SN6	60	20	✓	DDoS
SN6.1	60	20-120	-	low-rate DDoS
SN6.2	60	120-20	-	high-rate DDoS
SN7	500	100	✓	DDoS
SN7.1	500	10-200	-	low-rate DDoS
SN7.2	500	100-40	-	high-rate DDoS
SN7.3	500	30-50	-	low-rate DDoS

4.2 Results

The obtained results for all the scenarios are given in Table 2. This Section firstly refers to the DoS attack scenarios and then to DDoS ones. As shown in Table 2, we use legacy intrusion detection metrics, namely False Positive (FP) and False Negative (FN) to assess the performance of each algorithm. One can easily observe that in the case of DoS involving scenarios SN1.1 to SN5.2, the maximum FP value is equal to 3.7%, scored by both SMO and Neural Networks detectors. For the same scenarios, the FN metric remains low, presenting an average value of 0.02% and a maximum one of 0.85%. Generally, the best results in the DoS case are obtained by J48 and Random Forest classifiers. The results also indicate that as the attack traffic volume increases the FP and FN rates decrease. For instance, taking SN3.1 and SN4.1 as an example, the FP metric decreases significantly when compared to the first three subscenarios, namely SN1.1-SN1.3.

On the downside, the false alarms per classifier augment for scenarios SN6.1 to SN7.3 representing a DDoS case. This is rather expected as the occurrences per message header decrease significantly due to the multiple spoofed IPs - that in turn affect headers S3, S5 and S6 of virtually every transmitted SIP message - thus leading to a more difficult separation between the attack and normal messages.

Among all the classifiers the worst results for DDoS scenarios in terms of FP are obtained by SMO and Naive Bayes. Note that FP percentage rates scored in DDoS scenarios for all the algorithms are generally considerably higher than those obtained by the corresponding DoS ones. Taking SN6.1 for example, FP fluctuates between 0.04% and 17.7%, having an average value of 6.86%. Similar results are recorded for SN7.1, with FP varying between 5.2% and 11.3%. When the attack traffic increases, i.e., when the high-rate DDoS scenarios are involved, all the results are improved significantly. This is because the portion of the attack messages inside the same M_w increases proportionally to the rate of the attack. For instance, for scenario SN6.2, the maximum FP value is rather negligible, equal to 0.55%, while FN is zeroed. Similar results are obtained in the case of the other high-rate DDoS scenario, namely SN7.2, demonstrating a maximum FP value equal to 1%. Finally, SN7.3 corresponds to a moderate attack rate scenario and presents similar results to the four previously mentioned ones.

Specifically for DDoS scenarios, we compare the results scored by ML detectors against those obtained for the same scenarios but with two other anomaly-detection methods, namely Entropy (Shannon, 2001) and Hellinger Distance (Nikulin, 2001; Tsiatsikas et al., 2014). Table 3 summarizes the FP and FN results obtained by the two aforementioned schemes. To help the reader compare between the various algorithms, the rightmost columns of the same Table contain the corresponding false alarm values as scored by the top ML-based performer. Bear in mind that in contrast to ML techniques the training scenarios (SN6 and SN7) used for Entropy and Hellinger Distance do not include attack traffic. This is sensible because non-machine learning approaches rely on deviations between the legitimate messages in order to compute the corresponding thresholds. If an examined message exceeds the predefined threshold, then the message is classified as abnormal.

We can safely argue that the non-machine learning schemes score worse results in comparison with ML-based ones. More precisely, in the case of Entropy metric and for all the five DDoS scenarios, the FP rate reaches the maximum value of 18.1%, while FN varies between 5.41% and 43.5% (and especially for the Entropy metric scores exceedingly high values for all the scenarios but one). Further, the FP for Hellinger Distance fluctuates between 1.8% to 36%. The maximum FN value for the two aforementioned methods is the same, equal to 43.1% perceived in both cases for scenario SN6.2.

To sum up, the results obtained from Table 3 imply that ML-based detectors outperform the non-machine learning ones especially in terms of FN, for all DDoS incidents. In fact, the same category of detectors are overall competitive, presenting high accuracy in DoS scenarios as well. This is because these schemes learn from a mixed traffic including both

normal and attack messages, and thus it is easier for them to separate between the two classes, even with slight differences in header occurrences.

In general, anomaly-detection schemes must cope with a number of issues (Gates and Taylor, 2007): (i) A considerable number of false alarms (especially false positives) is normally expected by most classifiers. In our case, this statement seems to be confirmed in its entirety for the Entropy and Hellinger Distance metrics. For the ML ones, we can assert that the same statement stands half-true for FP, and false for FN. Specifically, ML-based detection largely fails in the case of low-rate DDoS (except for Neural Networks, and partially for Decision Trees), but it is effective across all algorithms for high-rate DDoS. This however hardly comes as a surprise as low-rate attacks are generally much harder to detect. (ii) Acquiring attack-free data for training may be a problem. In our case, this point can be dealt with if a VoIP billing system is in place. This will allow the correct labeling of each message because these logs are supposed to be accurate and valid. (iii) Smart aggressors may try to elude detection by increasingly teaching a system to identify intrusive activity as legitimate. To tackle this third point, one can vary the M_w based on mid or long-term statistical observations regarding SIP traffic.

A last point to be emphasized is that in terms of complexity ML-based classifiers require a different and usually significant amount of time to build a model from the given training set. Note that this time does not include that needed to generate an .arff data file from the given log file. For instance, taking SN4 as an example the training process spans between 0.01 to 154.95 secs for all the classifiers when fed with a file containing 261k records of SIP messages.

5 Related Work

In this Section, we detail on the related work and more specifically on contributions discussing the applicability of ML-driven techniques in detecting security incidents in VoIP services employing SIP or other similar signaling protocol.

The work in (Akbar and Farooq, 2009) proposes the use of ML techniques to detect flooding attacks against SIP-based services. The authors build their model to only consider the first line of an SIP message (S1 in our case), and ignoring the rest of the headers. They analyse the role of several classifiers and their effectiveness in detecting SIP flooding attacks. They assert that the false alarms produced are negligible. However, they take into account only DoS events and they create their datasets by artificially injecting the simulated attack traffic to the normal one. In opposite to that, we use realistically simulated attacks by using a rich variety of call rates and considering different configurations in the number of users.

The authors in (Akbar and Farooq, 2014), (Nassar et al., 2008) present two rather similar methodologies for protecting VoIP services against flooding and Spam over Internet Telephony (SPIT) attacks. For the first one, the authors introduce a real-time mechanism containing a feature computation module that extracts a set of spatial (changes in IDs or IP address) and temporal (call ratio) features from SIP packets. The generated vectors of features are fed to Naive Bayes and J48 classifiers. As in (Akbar and Farooq, 2009), for creating their training dataset, the authors inject the attacks into the normal traffic.

The work in (Nassar et al., 2008) proposes a real-time monitoring system to detect abnormal SIP messages based on SVM classifier. The authors make use of 38 dissimilar features aiming to detect SPIT and flooding assaults. These features are extracted by dividing SIP signalling into a number of small portions. A major difference from our work lies in the excessive number of features they use, which in turn may cause ambiguity in the classification process. Also, the authors concentrate solely on SVM, thus leaving aside sereral other ML detectors.

Lastly, the authors in (Bouzida and Mangin, 2008) introduce another framework for detecting anomalies in SIP. Their proposal capitalizes on the decision tree classifier, focusing on resource flooding attacks in general and password guessing ones in particular. They construct a model based on SIP attributes included in <To>, <From>, and <Username> headers. A detection accuracy of over 99% is reported. Nevertheless, this result refers to DoS incidents, while DDoS are left unaddressed.

6 Conclusions

In network intrusion detection, a typical method for exposing attacks is by tracking the network activity for any anomaly. That is, any discrepancy from a previously learned normal profile is identified as suspicious. This procedure is usually done using methods borrowed from the machine learning realm. So far, this potential have been examined in the literature in a great extend. However, as discussed in Section 5 in the case of VoIP in general and SIP in particular, works on this topic are not only scarce but also incomplete. To fill this striking gap, in this paper, we try to better assess the power of ML-based

Table 2: Summary of results for all the scenarios (The best performer per scenario in terms of FP is in bold).

SN	Traffic (Calls)		SMO		Naive Bayes		Neural Networks		Decision Trees (J48)		Random Forest	
			FP	FN	FP	FN	FP	FN	FP	FN	FP	FN
	Total Rec.	Attack Rec.	%	%	%	%	%	%	%	%	%	%
SN1.1	11.3k	9.7k	2.1	0	0.3	0	2	0	**0**	0	**0**	0
SN1.2	14k	12.3k	1.8	0	0.15	0	1.7	0	**0**	0	**0**	0
SN1.3	15.4k	11.3k	3.7	0	0.24	0	3.7	0	**0**	0	**0**	0
SN2.1	12k	7.9k	0.01	0	0.25	0	**0**	0	**0**	0	**0**	0
SN2.2	13k	9.2k	0.06	0	0.28	0	**0**	0	**0**	0	**0**	0
SN2.3	24.5k	22.8k	**0**	0	0.11	0	**0**	0	**0**	0	**0**	0
SN3.1	667k	568k	0.09	0.04	0.02	0.85	0.3	0.01	**0**	0.01	0.04	0.01
SN4.1	178k	168k	**0**	0.01	**0.01**	0.01	**0**	0	**0**	0	0.02	0
SN5.1	262k	200k	0.02	0.05	0.05	0.08	0.4	0.01	0.01	0.01	0.06	0.01
SN5.2	667k	611k	0.02	0.01	0.09	0.01	0.28	0.01	0.03	0.01	0.08	0.01
SN6.1	175k	23k	17.7	0.01	11.2	0	0.04	0	2.4	0	3	0
SN6.2	114k	50k	0.18	0	0.55	0	0.01	0	**0**	0	**0**	0
SN7.1	203k	11k	10.4	0	11.3	0	5.2	0	7.3	0	5.2	0
SN7.2	144k	50k	0.51	0	1	0	0.25	0	0.27	0	0.25	0
SN7.3	128k	33k	0.78	0	0.91	0	0.25	0	0.31	0	0.24	0

Table 3: Summary of evaluation metrics for Statistical Schemes in DDoS scenarios ($M_w = 1,000$).

SN	Low-rate	Entropy		Hellinger Distance		ML Techniques (Top performer)	
		FP	FN	FP	FN	FP	FN
		%	%	%	%	%	%
SN6.1	✓	**0**	13.3	36	0.01	0.04	0
SN6.2		0.97	43.5	1.8	0	**0**	0
SN7.1	✓	**4.4**	5.41	8	5.41	5.2	0
SN7.2		18.1	34.5	3.38	0	**0.25**	0
SN7.3	✓	**0**	25.7	2.49	5.45	0.24	0

techniques to identify (D)DoS incidents that capitalize on the use of SIP signaling. We consider 5 different popular ML detectors and a plethora of realistically simulated SIP traffic scenarios representing different flavors of (D)DoS. The results indicate that specific classifiers present high accuracy even in cases of low-rate DoS attacks. The best results for DDOS are obtained for the classifier introducing the maximum overhead, and thus accuracy may at a hefty price. To grab a better understanding of the effectiveness of this kind of detection, we compare the obtained results against those generated by two other anomaly-based methods, namely Entropy and Hellinger Distance. From this comparison one can safely argue that ML techniques appreciably surpass non-machine learning ones in terms of FN and up to a certain extend in terms of FP.

From the discussion given in the results subsection, one can mark down some directions for future work. The first one has to do with the extension of this work to embrace real-time detection of (D)DoS incidents using the same techniques. A second one involves extensive experimentation with the M_w parameter in an effort to better assess its overall effect on the detection process. The last one pertains to the evaluation of more advanced classifiers regarding its ability to cope with DDoS attacks in VoIP ecosystems.

Acknowledgements

This paper is part of the 5179 (SCYPE) research project, implemented within the context of the Greek Ministry of Development-General Secretariat of Research and Technology funded program Excellence II / Aristeia II, co-financed by the European Union/European Social Fund - Operational program Education and Life-long Learning and National funds.

REFERENCES

Akbar, M. A. and Farooq, M. (2009). Application of evolutionary algorithms in detection of sip based flooding attacks. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 1419–1426. ACM.

Akbar, M. A. and Farooq, M. (2014). Securing sip-based voip infrastructure against flooding attacks and spam over ip telephony. *Knowledge and information systems*, 38(2):491–510.

Bouzida, Y. and Mangin, C. (2008). A framework for detecting anomalies in voip networks. In *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on*, pages 204–211. IEEE.

Eastlake, D. and Hansen, T. (2011). Us secure hash algorithms (sha and sha-based hmac and hkdf). Technical report, RFC 6234, May.

Ehlert, S., Geneiatakis, D., and Magedanz, T. (2010). Survey of network security systems to counter sip-based denial-of-service attacks. *Computers and Security*, 29(2):225 – 243.

Gates, C. and Taylor, C. (2007). Challenging the anomaly detection paradigm: A provocative discussion. In *Proceedings of the 2006 Workshop on New Security Paradigms*, NSPW '06, pages 21–29, New York, NY, USA. ACM.

Geneiatakis, D., Dagiuklas, T., Kambourakis, G., Lambrinoudakis, C., Gritzalis, S., Ehlert, K., and Sisalem, D. (2006). Survey of security vulnerabilities in session initiation protocol. *Communications Surveys Tutorials, IEEE*, 8(3):68–81.

Geneiatakis, D., Kambourakis, G., Lambrinoudakis, C., Dagiuklas, T., and Gritzalis, S. (2005). Sip message tampering: The sql code injection attack. In *Proceedings of 13th International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2005), Split, Croatia*.

Geneiatakis, D., Kambourakis, G., Lambrinoudakis, C., Dagiuklas, T., and Gritzalis, S. (2007). A framework for protecting a sip-based infrastructure against malformed message attacks. *Communications Networks, Elsevier*, 51(10):2580–2593.

Geneiatakis, D., Vrakas, N., and Lambrinoudakis, C. (2009). Utilizing bloom filters for detecting flooding attacks against SIP based services. *Computers & Security*, 28(7):578–591.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18.

Kamailio (2014). the open source sip server. `http://www.kamailio.org/w/`.

Kambourakis, G., Kolias, C., Gritzalis, S., and Park, J. H. (2011). Dos attacks exploiting signaling in UMTS and IMS. *Computer Communications*, 34(3):226 – 235.

Keromytis, A. D. (2011). *Voice over IP Security - A Comprehensive Survey of Vulnerabilities and Academic Research.*, volume 1 of *Springer Briefs in Computer Science*. Springer.

Keromytis, A. D. (2012). A comprehensive survey of voice over ip security research. *IEEE Communications Surveys and Tutorials*, 14(2):514–537.

Krishnamurthy, R. and Rouskas, G. (2013). Evaluation of sip proxy server performance: Packet-level measurements and queuing model. In *Communications (ICC), 2013 IEEE International Conference on*, pages 2326–2330.

Mohr, C. (2014). Report: Global voip services market to reach 137 billion by 2020.

Nassar, M., Festor, O., et al. (2008). Monitoring sip traffic using support vector machines. In *Recent Advances in Intrusion Detection*, pages 311–330. Springer.

Nikulin, M. (2001). *Hellinger distance*. Encyclopeadia of Mathematics.

Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E. (2002). Sip: Session initiation protocol. Internet Requests for Comments.

Shannon, C. E. (2001). A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(1):3–55.

Stanek, J. and Kencl, L. (2011). Sipp-dd: Sip ddos flood-attack simulation tool. In *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*, pages 1–7.

Tang, J., Cheng, Y., Hao, Y., and Song, W. (2014). Sip flooding attack detection with a multi-dimensional sketch design. *Dependable and Secure Computing, IEEE Transactions on*, PP(99):1–1.

Tsiatsikas, Z., Geneiatakis, D., Kambourakis, G., and Keromytis, A. D. (2015). An efficient and easily deployable method for dealing with dos in sip services. *Computer Communications*, 57(0):50 – 63.

Tsiatsikas, Z., Kambourakis, G., and Geneiatakis, D. (2014). Exposing resource consumption attacks in internet multimedia services. In *proceedings of 14th IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, Security Track, pages 1–6. IEEE Press.

Witten, I. H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.