Under consideration for publication in Knowledge and Information Systems

Securing SIP based VoIP Infrastructure Against Flooding Attacks and Spam Over IP Telephony

Muhammad Ali Akbar and Muddassar Farooq

Next Generation Intelligent Networks Research Center (nexGIN RC)
National University of Computer & Emerging Sciences (FAST-NUCES)
Islamabad, Pakistan
Email: {ali.akbar, muddassar.farooq}@nexginrc.org

Abstract. Security of Session Initiation Protocol (SIP) servers is a serious concern of Voice over Internet (VoIP) vendors. The important contribution of our paper is an accurate and realtime attack classification system that detects: (1) application layer SIP flood attacks that result in Denial of Service (DoS) and Distributed DoS (DDoS) attacks, and (2) Spam over Internet Telephony (SPIT). The major advantage of our framework over existing schemes is that it performs packet-based analysis using a set of spatial and temporal features. As a result, we do not need to transform network packet streams into traffic flows, and thus save significant processing and memory overheads associated with the flow-based analysis. We evaluate our framework on a real-world SIP traffic – collected from the SIP server of a VoIP vendor – by injecting a number of application layer anomalies in it. The results of our experiments show that our proposed framework achieves significantly greater detection accuracy compared with existing state-of-the-art flooding and SPIT detection schemes.

Keywords: SIP; Intrusion detection; VoIP security; SPAM over IP Telephony; Denial of Service

1. Introduction

The global communication market is rapidly moving towards Voice over Internet commonly known as VoIP or Internet telephony. The prime motivation behind

Received Mar 04, 2012
Revised Jul 23, 2012
Accepted Oct 28, 2012

this trend is ubiquitous availability of high-bandwidth Internet at much cheaper rates compared with circuit switched telecommunication networks. In 2007, a German company, Ipoque, carried out an in-depth analysis of 3 petabytes of Internet traffic collected from five regions of the world. The results of their study show that 30% of German Internet users subscribe to the VoIP services (Ipoque, 2007). Similarly, market surveys in 2008 showed that VoIP accounts for 49.7% of total voice traffic at the end of year 2007 (The-VoIP-Network, 2008).

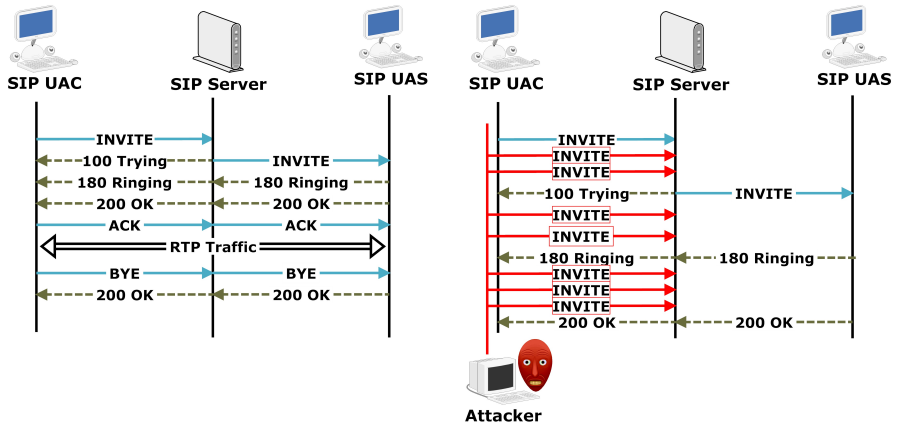
A reliable VoIP infrastructure must guarantee at least 99.9% uptime to stay competitive in the telecommunication market. Moreover, with an ever increasing popularity and penetration of Internet telephony, SIP servers, are becoming a hot target for imposters or intruders.(Remember that SIP protocol is the de facto standard for session signaling in VoIP.) A recent survey by SANS Institute supports our argument by suggesting that SIP servers are among the **SANS Top 20 Security Risks** (SANS-Institute, 2007). The intruders know that a successfully launched Denial of Service (DoS) attack on a SIP server can result not only in huge financial losses to the operator and its customers, but also seriously undermines its credibility. We, therefore, argue that reliable and robust SIP servers are at the core of any reliable VoIP infrastructure.

The major contribution of our paper is a dependable and secure framework for SIP servers that protects them against well-known application layer attacks. Our framework PbSIP (Packet-based SIP Intrusion Protector) consists of three modules: (1) packet analyzer, (2) feature computation, and (3) attack classification. The packet analyzer monitors the packets in the incoming traffic. (Remember our framework is purely packet-based, which eliminates the need to transform packets into flows.) The feature computation module computes a set of spatial and temporal features on SIP packets that are already processed by packet analyzer and placed in the buffer. Finally, the feature vector computed by the feature computation module is given as an input to the attack classification module – consisting of standard machine learning algorithms – that detects anomalous traffic on the basis of computed features. In case of an anomalous activity, it also raises an alarm. The results of our experiments show that our framework – using spatial and temporal features – successfully detects above-mentioned attacks with high detection accuracy in realtime during the signaling phase. This helps in taking effective countermeasures by filtering or blocking malicious traffic.

To summarize, our proposed SIP attack detection framework has following security features:

1. **Detection of application level DoS & DDoS flood attacks.** It successfully detects single source DoS flood attacks. Moreover, it also detects distributed DoS flood attacks launched from multiple sources.
2. **SPIT Detection.** It can detect social threats like SPAM over Internet Telephony (SPIT) in real-time during the signaling process. It is a challenge to detect SPIT during the signaling process because a spam call uses benign signaling mechanism.
3. **Robust mechanism of raising alarm.** It raises an alarm with relatively high accuracy on the basis of short and long time monitoring of SIP traffic.

The rest of the paper is organized as following. We present our threat model in Section 2. We present characteristics of our real-world VoIP dataset in Section 3 and describe our attack injection process in the collected benign traffic. In Section 4, we propose generic architecture of our security framework. We briefly



(a) SIP Signaling Model (b) INVITE Flood Attack

Fig. 1. Model of SIP calls

describe related work in Section 5 before discussing the results of our experiments in Section 6. We highlight important shortcoming of our framework in Section 7. Finally, we conclude the paper with an outlook to our future work.

2. Threat Model

The authors of (McGann and Sicker, 2005) and (Sisalem, Kuthan, Ehlert and Fokus, 2006) have provided a list of known VoIP vulnerabilities. Our proposed framework detects three types of attacks: (1) flooding attacks from single source that cause denial of service (DoS), (2) flooding attacks from multiple sources (DDoS), and (3) social attacks like SPAM over Internet telephony (SPIT). To make the paper self contained, we first describe the signaling model of a benign SIP call. Afterwards, we show how an imposter can exploit different vulnerabilities in the signaling model of SIP to launch different types of attacks.

2.1. SIP Call Model

Figure 1(a) shows the signaling model of a benign SIP call. We see that a SIP client follows the SIP protocol specification to complete the process of call establishment and call termination. In contrast, an intruder – by simply ignoring the SIP protocol specification – can launch INVITE flood attack (Figure 1(b)).

2.2. Flooding Attack for SIP

The first important threat to a SIP server, as mentioned before, is that an intruder can exploit the vulnerability in the signaling by initiating a very large number of SIP sessions from single or multiple IP addresses (Nassar, State and

Festor, 2008). The objective is to exhaust resources of a SIP server which it reserves for a SIP client; consequently, it results in denial of service to legitimate users. Our model takes care of two types of flooding attacks: (1) DoS attacks, and (2) Distributed DoS attacks.

DoS Attacks. This attack results due to brute force flooding of INVITE messages from a single IP address. Such attacks can be avoided by monitoring the frequency of packets arriving within a fixed time-window.

DDoS Attacks. Distributed DoS (DDoS) flood attacks are a variant of DoS attacks, in which an intruder first infects large number of hosts called reflectors. Afterwards, the attacking node sends its commands to the reflector. The reflector then launches small INVITE floods to the victim SIP server and forwards the command to other reflectors. The problem with this scenario is that the number of INVITE floods, generated by each reflector, is very small that makes it very difficult to differentiate between the malicious traffic of a reflector and that of benign SIP client. But the sum of INVITE packets from all reflectors overwhelms the victim SIP server (Sisalem et al., 2006), and as a result, it is unable to provide service to legitimate clients (Sisalem et al., 2006).

In this paper, we focus on INVITE flood attacks only as compared to other types of SIP flood e.g. REGISTER flood. As we are developing a SIP firewall for a real world SIP server, we have used this practical option for three reasons. First, the overhead for an incomplete INVITE handshake is significantly greater as compared to other SIP floods. Secondly, an INVITE attack affects two networks/servers as compared to the case of REGISTER floods that affects only a single network/server. The third and the most important reason for this choice is the billing model. VoIP vendors bill their clients (VoIP Application Servers) on the volume of calls (measured by INVITE packets). If a client uses a significant amount of bandwidth (number of concurrent calls) of the application server without paying for any of the incomplete calls, the company running the application server will suffer significant revenue loss. REGISTER flood attacks don't result in that much revenue loss; therefore, simple solutions such as rate limiting can be employed to reduce the threat of such attacks.

2.3. SPAM over IP Telephony

VoIP customers nowadays get a large number of unsolicited calls – known as Spam over IP Telephony (SPIT). Each spam in VoIP is a legitimate call request for a SIP server that forwards it to the called person. Once he picks up the telephone, only then he realizes that the call is a SPAM. If it occurs frequently or as a flood in a worst case scenario (Quittek, Niccolini, Tartarelli and Schlegel, 2006), it simply distracts a person from his main job which adds to to his frustration. SPIT is becoming an attractive and cost efficient marketing strategy because VoIP allows cheap calls compared with traditional PSTN networks. Almost all of existing solutions like *source filtering, white listing, grey listing* and *handshake challenge* require some form of user intervention (Radermacher, 2005). To the best of our knowledge, no accurate solution exists that detects a SPAM during the call establishment process. Our framework provides protection against following attacks: (1) DoS and DDoS flooding, and (2) SPIT.

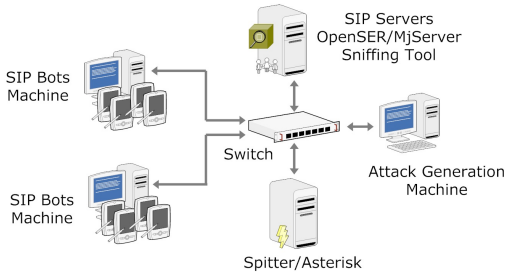


Fig. 2. Testbed for attack generation

3. SIP Dataset

We believe that collecting a real-world SIP dataset is very important in validating our proposed security framework. To achieve this important objective, we contacted a VoIP vendor that has a customer base in North America, but houses its VoIP infrastructure in Pakistan. (The name of the company is kept anonymous as 'FRIENDS' at the request of CTO of the company.) We developed a SIP traffic logger and deployed it on the SIP server of FRIENDS. Our dataset consists of a log of more than 20 days. The traffic set mostly contains low calls rate except for the peak hours. Therefore, we decided to filter out the peak hours traffic, rank it and use top peak hour traffic in our experiments. We now discuss that how we inject anomalies into benign dataset.

3.1. Malicious Data Set

We have assembled a testbed for generation of SIP attacks. The testbed is shown in Figure 2. The testbed consists of opensource SIP servers (OpenSER and MjServer), a spam generation opensource tool (Spitter running on top of Asterisk), attack generation machine which is capable of launching DoS and DDoS attacks using SIP bots and also controls Spitter/Asterisk for spam generation.

We have developed a tool that injects different types of SIP anomalies in collected benign traffic sets. The tool has the capability to inject anomalies both in time and space (see Table 1 and Table 2). For temporal anomalies, we inject small and long duration attacks. We define small duration attacks of 30 seconds as "harmonic attacks". Similarly, long duration attacks of 2 minutes and 10 minutes are defined as "chunk attacks". Harmonic attacks represent an attacker who periodically launches small duration attacks. The aim is to evade detection that might result in case of an attack of longer duration. In a chunk attack, an intruder constantly floods a SIP server for a longer period of time to avoid creating abrupt transients in the traffic pattern – making its detection difficult.

We can also launch an attack from a single source or from a number of multiple sources. These attacks show variation in IP address space and hence will test the ability of a classification system to distinguish between DoS and DDoS attacks. Finally, we compliment both types of attack with a third dimension of intensity by varying the rate of INVITE packet floods from very low to very high. Very low (VL), Low (L), Medium (M), High (H) and Very High (VH) means a flood of 10 packet/sec, 25 packets/sec, 50 packets/sec, 100 packets/sec and 500 packets/sec respectively.

Table 1. Flooding Attack Scenarios: VL, L, M, H, VH means 10, 25, 50, 100, 500 packet/sec of attack rate respectively.

Scenario	Type	Duration	Attack source(s)	Intensities
Scenario 1	Harmonic	30 sec	single	VL,L,M,H,VH
Scenario 2	Harmonic	30 sec	multiple	VL,L,M,H,VH
Scenario 3	Chunk	2 min	single	VL,L,M,H,VH
Scenario 4	Chunk	2 min	multiple	VL,L,M,H,VH
Scenario 5	Chunk	10 min	single	VL,L,M,H,VH
Scenario 6	Chunk	10 min	multiple	VL,L,M,H,VH

Table 2. SPAM generation scenarios

Scenario	Duration of Attack	Concurrent call rate	Avg SPAM Call Duration	Hit Rate %
Scenario 1	2min	1,5,20,100 cps	10sec	10
Scenario 2	2min	1,5,20,100 cps	10sec	100

In order to create challenging attack scenarios, we systematically inject a combination of above-mentioned attacks to analyze the detection behavior of a classification system. As a result, we have created six different attacks scenarios as shown in Table 1. We believe that these six scenarios cover almost every possible type of flood combination.

3.1.1. Anomalies injection for SPIT

We generate different scenarios of SPAM attacks in our testbed. The spam calls are generated using Spitter/Asterisk tool and are received by the SIP bots which either pick the call or send an error message (usually `486 Busy` or `302 Moved`). We consider two different scenarios. In the first scenario, the hit rate (spam calls to existent bots) is 10% and in the the second scenario, the hit rate is 100%. For each scenario, we generate spam calls with varying intensities (1, 5, 20 and 100 concurrent spam calls). Table 2 shows the SPAM attack scenarios that are used in the experiments.

4. Architecture of PbSIP

Our framework consists of three modules: (1) packet analyzer, (2) feature computation, and (3) attack classification. The packet analyzer monitors the packets in the incoming traffic. (Remember our framework is purely packet-based, which eliminates the need to transform packets into flows.) The feature computation module computes a set of spatial and temporal features on SIP packets that are processed by packet analyzer and placed in the buffer. At the moment we compute features on a window of x (currently 40) packets. Finally, the feature vector computed by the feature computation module is given as an input to the Naive Bayes classifier, which detects anomalous traffic on the basis of computed

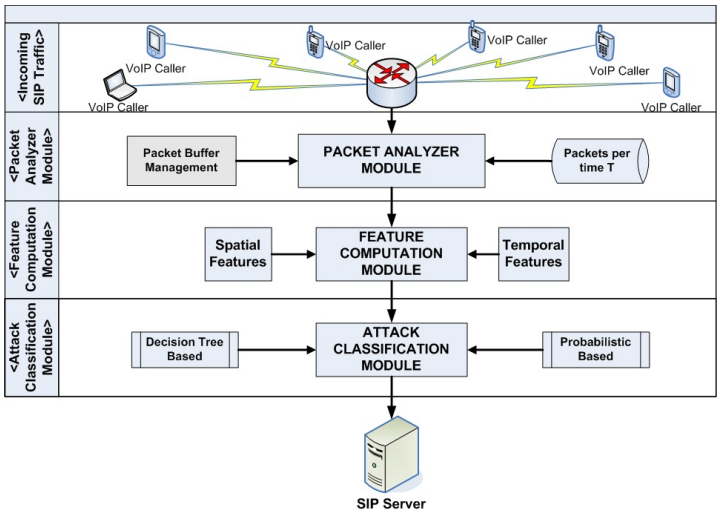


Fig. 3. Architecture of PbSIP

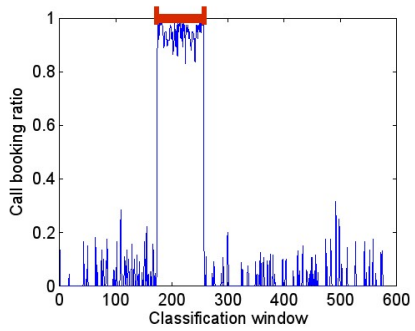
features. In case of an anomalous activity, it also raises an alarm. The results of our experiments show that our framework – using spatial and temporal features – successfully detects: (1) INVITE based DoS, (2) INVITE based DDoS, and (3) SPIT. All these attacks are detected in realtime, which helps in taking effective countermeasures by filtering or blocking malicious traffic. We now discuss details of each module.

4.1. Packet Analyzer Module

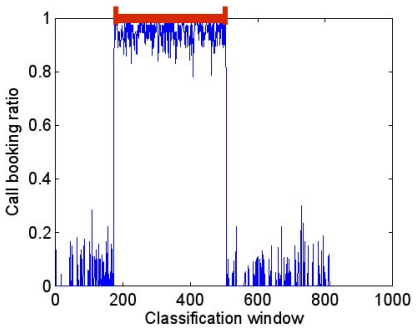
The job of packet analyzer is to transform a network packets' stream into the format supported by the feature computation module. We have two design options for this module: (1) flow based monitoring, and (2) packet based monitoring. The flow based monitoring is mostly used by the network attack detection systems, which operate on offline SNMP data or flow logs generated by routers (such as NetFlow or CFlow). The flow based monitors have limited utility at the application layer of endpoints because of high processing and memory overheads in maintaining and generating flow records. Packet based monitoring, on the other hand, is based on extracting the forensic information from the packet's header and data. Packet based monitoring not only ensures minimum latency in attack detection but also small processing and memory overheads (*Packet vs flow-based anomaly detection*, n.d.). After carefully analyzing merits/demerits of both approaches, we have decided to opt for packet based monitoring in PbSIP.

4.2. Feature Computation Module

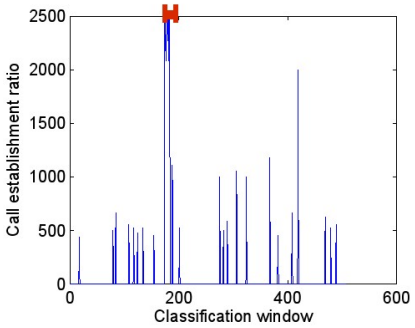
The feature computation module in PbSIP consists of two cascaded blocks: (1) the data structures for storing the extracted information from SIP packets, and (2) the features' processor that computes relevant features from the stored infor-



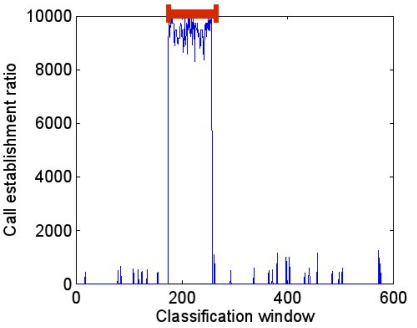
(a) Call booking ratio for Harmonic Attack



(b) Call booking ratio for Chunk Attack



(c) Call establishment ratio for low intensity flood



(d) Call establishment ratio for high intensity flood

Fig. 4. Analysis of temporal features – Red bar shows attack interval

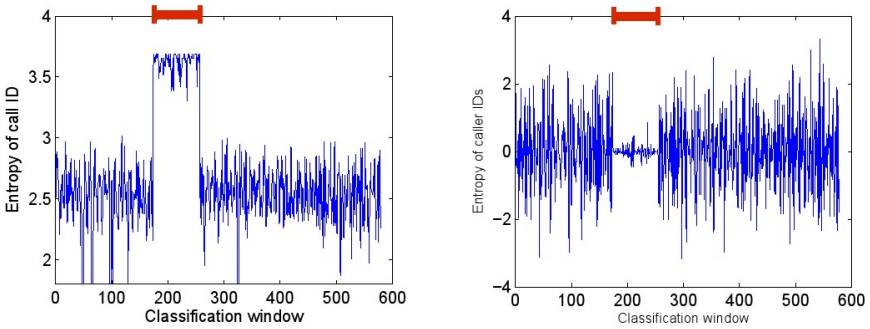
mation. We use efficient data structures like *maps*, which have small processing overheads for information storage and retrieval.

In the next step, PbSIP uses a set of spatial and temporal SIP traffic features for attack detection. We have identified two temporal and two spatial features for detection of flooding attacks. The criteria for feature selection are that they: (1) have the potential to discriminate between benign and anomalous traffic patterns, (2) reduce the dimensionality of the input feature space without making a compromise on detection accuracy, (3) have small memory and processing overheads that do not explode with respect to the volume of observed traffic.

4.2.1. Flooding Attacks.

Now we present our two temporal and two spatial traffic features for flood attacks.

Temporal features for flood attacks. The temporal traffic features are designed to capture the temporal information in SIP traffic. The module com-



(a) Entropy of Call IDs for DoS attack

(b) Entropy of Caller IDs for DDoS attack

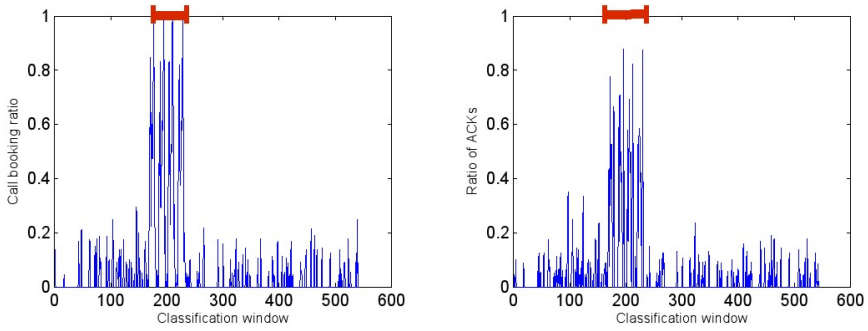
Fig. 5. Analysis of Spatial features – Red bar shows attack interval

putes them in realtime directly on the incoming stream of SIP packets. Recall from Figure 1 that a user generally initiates three types of request messages INVITE, ACK and BYE. We have selected two temporal features: (1) call booking ratio, and (2) call establishment ratio. The call booking ratio is the ratio between the number of INVITE packets in a window to the total number of request packets. We also define call establishment ratio as the ratio of the number of INVITE packets in a window to the number of ACK packets. This feature indicates that how many calls which are initiated are successfully established. We argue that call booking ratio will increase during an INVITE flood because the rate of other request messages like REGISTER, BYE, OPTIONS, CANCEL, UPDATE, REFER, SUBSCRIBE, NOTIFY, MESSAGE, INFO and PRACK remains unchanged. Similarly, an attacker will have no interest in completing a call by sending an ACK; as a result, call establishment ratio will also increase.

In order to confirm our hypothesis, we plot both ratios in Figure 4. It is clear from Figures 4(a) and 4(b) that call booking ratio, as expected, increases during harmonic and chunk INVITE floods. Similarly call establishment ratio also increases during low and high attack rates as is evident in Figures 4(c) and 4(d).

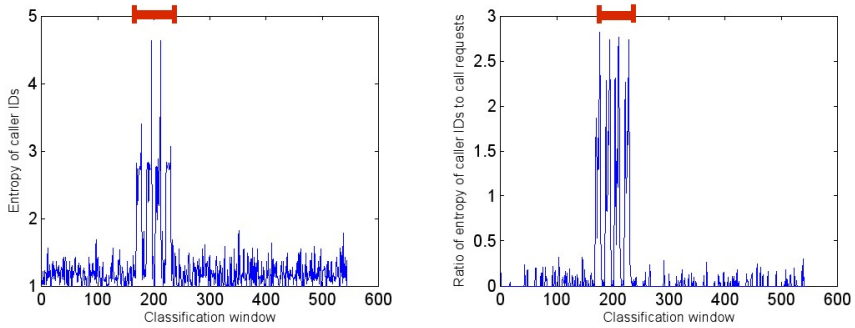
Spatial features for flood attacks. The spatial traffic features are designed to capture information about changes in address or IDs. Our intuition is that during INVITE floods the number of caller IDs and the number of call IDs will get perturbed. The reason for this is that an intruder in DoS attack will try to artificially initiate a large number of calls by varying only their IDs and in case of DDoS attack callers on different hosts will try to initiate simultaneously large number of calls.

To achieve this objective, we use Entropy that is an information-theoretic measure to capture the increased variance in caller IDs and call IDs in case of DoS or DDoS attacks. We have accordingly defined two features to capture the



(a) Call booking ratio

(b) Ratio of ACKs



(c) Entropy of caller IDs

(d) Ratio of entropy of caller IDs to call requests

Fig. 6. Perturbation of features for SPIT (Red line marks the attack period)

variance: (1) entropy of Caller IDs, and (2) entropy of Call IDs. We calculate entropy using the following formula:

$$H_n = -\sum_{i=1}^{k} \frac{c_i}{k} \log_2(\frac{c_i}{k}) \tag{1}$$

where H_n is the entropy of Caller ID in the n_{th} classification window, k is the window size and c_i is the frequency of packets from i_{th} Caller ID during the n_{th} window. Similarly, we compute entropy of Call ID. We argue that the value of both of these features during an attack will deviate from its normal value. Figure 5 confirms our hypothesis.

4.2.2. Spam over Internet Telephony (SPIT)

In order to detect SPIT, we propose two temporal features and two spatial features.

Temporal features for SPAM detection. We argue that during SPAM call booking ratio and ratio of `ACK` packets will increase. Figures 6(a) and 6(b) confirm our point of view.

Spatial features for SPAM detection. We argue that SPAM consists of a storm of short advertisement calls to a number of receivers by the same sender. We, therefore, should be able to detect it using entropy of caller IDs at a SIP server. Figure 6(c) confirms our observation. The entropy of caller IDs is calculated using Equation 1 by taking c_i as the frequency of packets from i_{th} caller ID during the n_{th} window. Moreover, we define another spatial feature which is obtained by taking the ratio of entropy of caller IDs to the call booking requests. Figure 6(d) shows the perturbation in this feature during a spam attack.

4.3. Attack Classification Module

Once the features are computed by the feature computation module, they are fed to the attack classification module. The well-known machine learning and data mining classifiers: Naive Bayes, decision trees, inductive rules, instances and support vector machines. Data mining algorithms have the potential to uncover anomalies within huge traffic scenarios (Pham, Saha, Phung and Venkatesh, 2012) (Branch, Giannella, Szymanski, Wolff and Kargupta, 2012) and have been extensively used for security & privacy frameworks (Yang, Sato and Nakagawa, 2011) (Gundecha, Barbier and Liu, 2011)(McCue, 2011). Our choice of a classifier is motivated by its detection accuracy, training and testing overheads. After doing some empirical studies, we have short listed Naive Bayes and J48 classifiers as potential candidates for our attack classification module. We provide them above-mentioned temporal and spatial features for detecting flooding and SPIT attacks. We provide a brief description of classifiers to make the paper self contained.

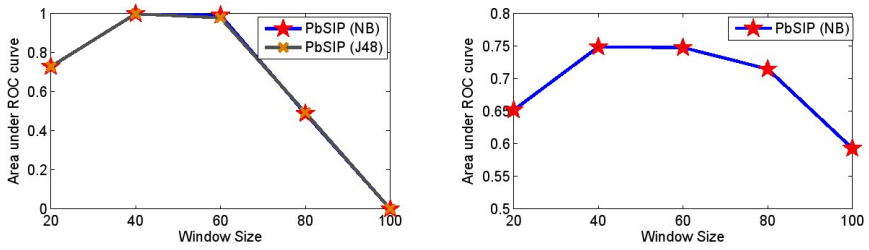
4.4. Naïve Bayes

Naïve Bayes is a simple probabilistic classifier assuming naïve independence among the features i.e. the presence or absence of a feature does not affect any other feature (Maron and Kuhns, 1960). The algorithm works effectively and efficiently when trained in a supervised learning environment. Due to its inherent simple structure it often gives very good performance in complex real world scenarios. The maximum likelihood technique is used for parameter estimation of Naïve Bayes models.

We use the default parameters for Naïve Bayes in WEKA (Witten and Frank, 2005). We neither use kernel estimator functions nor numeric attributes for supervised discretization that converts numeric attributes to nominal ones.

4.5. Decision Tree (J48)

Decision trees are usually used to map observations about an item to conclusions about the item's target value using some predictive models (Quinlan, 1993). They are very easy to understand and are efficient in terms of time especially on large datasets. They can be applied on both numerical and categorical data, and statistical validation of the results is also possible.



(a) Flood Attacks (Scenario 1 with 25 cps)

(b) Spam Attacks (Spam Rate = 20)

Fig. 7. Effect of variation in Window Size on Detection Accuracy (Represented by Area under ROC curve)

We use C4.5 decision tree (J48) that is implemented in WEKA. We use the default parameters for J48. We do not utilize binary splits on nominal attributes for building trees because all of our selected features are numeric. The confidence factor for pruning is set to 0.25, where lower values lead to more pruning. The minimum number of instance per leaf equals 2. The number of folds of training data is set to 3, where one fold is used for pruning and the rest are used for growing the tree.

4.6. Selection of Packet Window Size

The window size is an important parameter that might severely affect the detection accuracy of the system. If the chosen window size is too small, the framework will not be able to capture the benign behavior of SIP traffic for doing analysis of intrusions. If the chosen window size is too large, the low rate attacks might be masked in the large amount of traffic and hence might get skipped (or missed). Moreover, a larger window size also results in significant delays in detecting the attacks and taking associated countermeasures to prevent them. To conclude, it is relevant to select an appropriate window size.

We have done experiments to empirically evaluate the performance of PbSIP for both flooding and spam attacks by varying the window size between 20 to 100 packets. Figure 7 shows the impact of changing the window size on the detection accuracy that is represented by the area under (Rate of Convergence) ROC curve.

The results for Scenario 1, which consists of a very low call rate (25 cps, 30 sec Harmonic attack), are presented in Figure 7(a). Remember it is a challenging scenario; therefore, the effect of varying window size should be noticeable on the detection accuracy of the system. Similarly, we present the results for spam rate of 20 concurrent calls in Figure 7(b). It is evident from Figure 7 that PbSIP is able to discriminate *attack traffic* from *benign traffic* with a high degree of accuracy when the window size is in between 40 and 60 packets. As a result, we have selected a window size of 40 (the smallest in the best performance range).

5. Related Work

We now briefly describe the related work. This will help in developing better insights of the novel direction of our work compared with prior art. Both denial of service (DoS) attacks and Spam over Internet Telephony (SPIT) have been well known problems in the VoIP community for several years. The exponential increase in the volume of VoIP traffic has made VoIP infrastructure an attractive target for malicious intruders; therefore, a number of intrusion detection solutions for detecting DoS, DDoS attacks and spam have been proposed in the research literature.

The authors of (Geneiatakis, Vrakas and Lambrinoudakis, 2009) have proposed a hash based flooding detection mechanism for detecting flood attacks with a focus on avoiding end-to-end delays. Ormaza-bal et al. (Ormazabal, Nagpal, Yardeni and Schulzrinne, 2008) have designed a SIP firewall that uses rate-limiting to prevent DoS attacks. The authors in (Liu, 2011) have used Colored Petri Nets to achieve the same objective. Some variants of above-mentioned techniques have been used to detect flood attacks on a SIP server (Ehlert, Rebahi and Magedanz, 2009)(Akbar and Farooq, 2009)(Rafique, Ali Akbar and Farooq, 2009)(Thandeeswaran, Asha et al., 2012)(Chen, Wen and Yu, 2012). Spam detection on VoIP networks has received relatively less attention compared with flood attacks. Ono et al. (Ono and Schulzrinne, 2009) have utilized the concept of weak and strong social ties to detect SPIT. Wu et al. (Wu, Bagchi, Singh and Wita, 2009) have used a scalable semi supervised clustering technique to detect SPIT and evaluate their proposed scheme on synthetic traces. The closest approach to our work is taken by Nassar et al. (Nassar et al., 2008) who use features, extracted from SIP signaling, to train a support vector machine classifier and detect SPIT attacks. Other similar SPIT detection frameworks are presented in (Jung, Martin, Ernst and Leduc, 2012)(Sengar, Wang and Nichols, 2011)(Chaisamran, Okuda, Blanc and Yamaguchi, 2011). Keromytis (Keromytis, n.d.) has written a comprehensive survey on VoIP security that provides a detailed account of solutions to mitigate SIP flooding and SPIT attacks.

The authors of (Sengar, Wang, Wijesekera and Jajodia, 2008) have used Hellinger Distance (HD) to detect SIP flooding attacks. Their scheme measures the arrival rate of different types of SIP packets at a SIP server within a certain window of packets. HD uses the following formula to measure the distance between distribution of packets in benign and malicious datasets:

$$\begin{aligned}\texttt{HD} = (\sqrt{p_{\texttt{INVITE}}} - \sqrt{q_{\texttt{INVITE}}})^2 + (\sqrt{p_{\texttt{200OK}}} - \sqrt{q_{\texttt{200OK}}})^2 \\ +(\sqrt{p_{\texttt{ACK}}} - \sqrt{q_{\texttt{ACK}}})^2 + (\sqrt{p_{\texttt{BYE}}} - \sqrt{q_{\texttt{BYE}}})^2,\end{aligned}$$

where p and q represent the normalized frequencies of certain types of packets in benign and malicious datasets. In an earlier work reported in (Akbar, Tariq and Farooq, 2008), we have shown that HD provides the best detection accuracy in different types of flooding attack scenarios. However, it suffers from three shortcomings: (1) calculation of Hellinger distance requires supervised learning of p in the benign traffic; (2) a single temporal feature may not provide adequate protection against DDoS attacks; and (3) it cannot protect against SPIT. In comparison, we use a set of spatial and temporal features that provide protection against DoS, DDoS, and SPIT attacks. Hellinger Distance has been used by several authors for detecting SIP flooding attacks (Sengar, Wang, Wijesekera and Jajodia, 2006)(Sengar et al., 2008)(Tang, Cheng and Zhou, 2009)(Akbar

et al., 2008)(Kumar, Rahul and Joonuthula, 2011) because it provides relatively better accuracy even for low flood attack rates. As a consequence, we decided to compare our framework with HD to verify the claim: the presented technique provides approximately the same accuracy as compared with the Hellinger distance but is more resilient to evasion attempts because it uses a number of spatio-temporal features.

We also compare our scheme with F-SVM, which is recently published in (Nassar et al., 2008). Its features' set consists of 38 features on a per flow basis and it is given to a SVM classifier for eventual classification. (SVM classifier is a well-known classifier for supervised learning and has been extensively used in security and privacy (Vaidya, Yu and Jiang, 2008).) The results of our experiments show that F-SVM is able to successfully counter DoS and SPIT attacks. We argue that to compute 38 features in real time on a per flow basis will not only incur processing overhead but would also require relatively large memory for storing the per flow state. As a result, its utility – like classic flow based techniques – is only limited to doing an off-line forensic analysis of traffic logs to detect intrusions. In comparison, we compute a set of four spatial and temporal features on a stream of incoming packets and it will have less processing and memory overheads compared with that of F-SVM.

6. Experimental Results and Analysis

Recall from Section 3 that we have generated 6 different attack scenarios (see Section 3) just for flooding attacks. We believe that these scenarios cover a wide spectrum of "crafty attacks" that range from very low rate single source to very high rate multiple sources flood attacks. Moreover, the motivation for developing our framework is to have high classification accuracy of detecting different types of attacks with low processing and memory overheads compared with existing techniques. Our evaluation and validation framework is developed to empirically quantify these objectives. Therefore, we compare our approach with others using four performance metrics: (1) classification accuracy in terms of true positive rate (TP rate) and false positive rate (FP rate), (2) processing overheads of feature extraction module, and training/testing phases of a classifier, and (3) memory overheads (the size of buffers needed to store features).

We consider TP rate = $\frac{TP}{TP+FN}$, and FP rate = $\frac{FP}{FP+TN}$ (Fawcett, 2004). We use the terms *detection accuracy* and TP rate interchangeably in the paper. Similarly, *false alarm rate* and FP rate refer to the same thing. We discover the complete ROC space (Fawcett, 2004) of a given classifier and choose the best TP rate for which FP rate is less than 1%. The FP rate larger than 1% is not acceptable because it results in rejecting the calls of legitimate users. Ideally, we should have a zero FP rate to ensure that the calls of legitimate users are never rejected.

We first discuss the results of six flooding attack scenarios described in Table 1. Then we report the results for SPIT detection. Finally, we conclude the section with processing and memory overhead analysis of different schemes.

Table 3. Classification results for different variations of the INVITE flooding attacks (TP and FP rates in %)

Scenario 1 (Harmonic)										
			HD		F-SVM		PbSIP(J48)		PbSIP(NB)	
Type	Duration	Source	TP	FP	TP	FP	TP	FP	TP	FP
VL	30sec	DoS	100	0	75	0	92	0.2	92	0.2
L	30sec	DoS	100	0	95.7	0.2	96	0.2	96	0.2
M	30sec	DoS	100	0	97.6	0	98	0.2	98	0.2
H	30sec	DoS	100	0	98.7	0.4	99	0.4	99	0.4
VH	30sec	DoS	100	0	99.4	0.3	100	0.6	100	0.6
Scenario 2 (Harmonic)										
			HD		F-SVM		PbSIP(J48)		PbSIP(NB)	
Type	Duration	Source	TP	FP	TP	FP	TP	FP	TP	FP
VL	30sec	DDoS	100	0	75	0	91.7	0.2	91.6	0
L	30sec	DDoS	100	0	95.7	0.2	95.7	0.2	95.6	0.2
M	30sec	DDoS	100	0	97.6	0.2	97.6	0.2	97.6	0.2
H	30sec	DDoS	100	0	100	0.2	100	0.2	100	0.2
VH	30sec	DDoS	100	0	100	0.8	100	0.8	100	0.8
Scenario 3 (Chunk Attack)										
			HD		F-SVM		PbSIP(J48)		PbSIP(NB)	
Type	Duration	Source	TP	FP	TP	FP	TP	FP	TP	FP
VL	2min	DoS	100	0	76.6	0	98	0	98	0
L	2min	DoS	100	0	98	0	99	0	99	0
M	2min	DoS	100	0	98.8	0	99	0	99	0
H	2min	DoS	100	0	99.7	0.2	100	0.2	100	0.2
VH	2min	DoS	100	0	99.8	0.5	100	0.63	100	0.63
Scenario 4 (Chunk Attack)										
			HD		F-SVM		PbSIP(J48)		PbSIP(NB)	
Type	Duration	Source	TP	FP	TP	FP	TP	FP	TP	FP
VL	2min	DDoS	100	0	91.5	0	97.9	0.2	97.8	0
L	2min	DDoS	100	0	98.9	0	98.9	0.2	98.9	0
M	2min	DDoS	100	0	99.4	0.2	99.4	0.4	99.4	0.2
H	2min	DDoS	100	0	100	0.2	100	0.4	100	0.2
VH	2min	DDoS	100	0	100	0.6	100	0.6	100	0.6
Scenario 5 (Chunk Attack)										
			HD		F-SVM		PbSIP(J48)		PbSIP(NB)	
Type	Duration	Source	TP	FP	TP	FP	TP	FP	TP	FP
VL	10min	DoS	100	0	97.8	0.2	100	0.2	99.6	0.2
L	10min	DoS	100	0	99.8	0.2	100	0.2	99.8	0.2
M	10min	DoS	100	0	99.8	0.2	100	0.2	100	0.2
H	10min	DoS	100	0	99.9	0.4	100	0.5	100	0.5
VH	10min	DoS	100	0	99.9	0.7	100	0.9	100	0.7
Scenario 6 (Chunk Attack)										
			HD		F-SVM		PbSIP(J48)		PbSIP(NB)	
Type	Duration	Source	TP	FP	TP	FP	TP	FP	TP	FP
VL	10min	DDoS	100	0	98.3	0.2	99.6	0.2	99.5	0.2
L	10min	DDoS	100	0	99.8	0.2	99.8	0.2	99.7	0.2
M	10min	DDoS	100	0	99.9	0.2	99.9	0.2	99.8	0.2
H	10min	DDoS	100	0	100	0.2	100	0.2	100	0.2
VH	10min	DDoS	100	0	100	0.7	100	0.7	100	0.72

6.1. Detection of Flooding Attacks

Scenarios 1 and 2: Harmonic Attacks. The purpose of harmonic attacks is to study the behavior of different intrusion detection schemes when an intruder launches intermittently an attack of very small duration (30 seconds) by varying its intensity from very low (VL) to very high (VH). These attacks will provide an insight that how quickly a scheme detects an attack. Scenario 1 depicts an attack from the single source while scenario 2 represents a DDoS attack. As expected, very low attacks are relatively difficult to detect. It is obvious from Table 3 that HD is a benchmark algorithm that detects with 100% tp rate both types of harmonic attacks. F-SVM has poor detection rates for VL attacks in both scenarios. We investigated the problem and found that the attack rates of just 10 packets/sec do not perturb simple flow based features of F-SVM. Our proposed framework, PbSIP with NB classifier, successfully detects VL, L,M, H and VH rate attacks.

Scenarios 3 and 4: Chunk Attacks. The purpose of chunk attacks is to study the behavior of different intrusion detection schemes when an intruder launches a flood of relatively large duration (2 minutes) in Scenarios 3 and 4. Again we vary the attack rate from VL to VH. HD again shows excellent detection performance due to prior knowledge of benign traffic. In comparison, the accuracy of F-SVM shows poor performance for VL DoS attacks but its performance has significantly improved for VL DDoS attacks compared with harmonic attacks. This shows that simple features of F-SVM are suited to detect chunk DDoS attacks. PbSIP with NB, shows a consistent behavior by achieving a TP rate of 97.8% or above for all attack rates in both scenarios.

Scenarios 5 and 6: Chunk Attacks. We now investigate the effect of chunk attack if its duration is increased to 10 minutes. We see that all schemes are able to detect such prolonged attacks with high probability.

We can draw two important conclusions: (1) DDoS attacks are generally easier to detect compared with DoS attacks, and (2) attacks with VL attack rate and smaller duration are difficult to detect. We do acknowledge the fact that attack rates of 10 or 25 packets/sec are designed just to create a stress test for different schemes; otherwise, in real scenarios an attacker can not do significant damage with such small attack rates. To conclude, our PbSIP framework detects VL to VH attack rates with nearly same/better accuracy compared with existing state-of-the-art detection schemes *without the prior knowledge of benign traffic* and *less number of features resulting in lower memory and processing overheads.*

6.2. Detection of Spam over Internet telephony (SPIT)

We now focus our attention to SPIT problem. Recall that we use three temporal and 1 spatial feature to detect SPIT. We have already shown that during SPAM, values of these features are significantly perturbed. We have generated a spam of 1, 5, 20 and 100 concurrent calls. The average duration of these calls is 10 seconds. We have performed the experiments with two hit rates (10% and 100%). On the basis of the results in Table 3, we have decided to use Naive Bayes classifier. We compare our framework with HD and F-SVM on the same datasets.

We have tabulated the results of spam experiments in Table 4. It is clear from the results that HD has been designed only for the detection of flood attacks and it is unable to detect any spam call as the signaling process of spam calls

Table 4. Results of SPAM detection (TP and FP rates in %)

Scenario 1 (Spam Hit Rate = 10%)								
window size	Duration	Spam Rate	HD		F-SVM		PbSIP(NB)	
			TP	FP	TP	FP	TP	FP
40	2min	1	0	0	0	0	0	0
40	2min	5	0	0	0	0	31.6	0.8
40	2min	20	0	0	47.1	0	64.7	0.8
40	2min	100	0	0	73.5	0	95.1	0.8
Scenario 2 (Spam Hit Rate = 100%)								
window size	Duration	Spam Rate	HD		F-SVM		PbSIP(NB)	
			TP	FP	TP	FP	TP	FP
40	2min	1	0	0	0	0	0	0
40	2min	5	0	0	0	0	21	1
40	2min	20	0	0	54.2	0	70.8	0.4
40	2min	100	0	0	69.4	1	82.1	1

is similar to the normal calls. Results show that it is difficult to detect single spam calls. The trends of the results for both hit rates are similar. But it is easier to detect spam attacks with lower hit rate. PbSIP achieves significantly higher accuracy of detecting spam compared with F-SVM. PbSIP achieves 31.6% accuracy compared with 0% of F-SVM at a SPAM rate of just 5 calls with 10% hit rate. As we increase the spam rate, PbSIP continues to significantly outperform F-SVM. For example for a spam rate of 100 concurrent calls, PbSIP achieves 95.1% detection accuracy compared with just 73.5% of F-SVM for 10% hit rate. This shows that only a combination of spital and temporal features has the ability to detect SPIT during the signaling process. As expected, HD is unreliable in detecting SPIT calls.

6.2.1. Processing and Memory Overheads.

We report the processing and memory overheads of our PbSIP scheme in Table 5. In order to measure the processing overheads, we have used an in-execution realtime profiling tool. We divide the processing costs into three parts: (1) packet handling time, (2) feature calculation time, and (3) classification time. The packet handling cost is fixed for every scheme while feature calculation time is dependent on the number and complexity of features. Similarly, training and testing times of a classifier are dependent on its learning technique. We expect that our training and testing time will be significantly smaller compared with SVM because of simple learning behavior of Naive Bayes. In operations, testing time plays a critical role because we have to raise an alarm in realtime. We have conducted our experiments on a 1.8 GHz Intel Core 2 Duo processor with 2 MB L2 cache and 1 GB RAM.

It is clear from Table 5 that testing time of Naive Bayes is approximately four times smaller compared with SVM. The benefit of using packet based feature extractor is obvious in Table 5: the complexity of PbSIP is approximately 3 times smaller compared with F-SVM. Now if we consider the total processing overhead (T_o), F-SVM takes 88 msec compared with 25 msec of PbSIP. To conclude, if a SIP server is patched with PbSIP, then it can process more than twice the number

Table 5. Processing and memory overheads.

Overheads	F-SVM	PbSIP
Training Time (msec)	899.478	**16**
Packet Handling Time (T_p) (msec)	7.857	7.857
Feature Computation Time (T_f) (msec)	8.105	**2.714**
Testing Time (T_t) (msec)	72.928	**15**
$T_o = T_p + T_f + T_t$ **(msec)**	88.8	**25.571**
Memory (bytes)	304	**32**

of calls if it is patched with F-SVM. (We assume that in steady state the window of 40 packets is always full). This is of course a significant improvement. Similarly to store features of a packet stream, PbSIP requires just 32 bytes compared with 304 bytes of F-SVM.

7. Attacks and Countermeasures

In this section, we discuss the possible attacks against our proposed PbSIP framework. Adversaries can devise different ways to circumvent detection, if they know the detailed implementation of our framework. We discuss possible attacks for each module in our proposed framework.

Attacks on Traffic Monitoring Module. The traffic monitoring module in our framework may crash in case of a malformed packet given to it. In future, we want to filter malformed packets by applying grammar rules before parsing.

Attacks on Feature Computation Module. A crafty attacker can possibly circumvent monitoring of Call/Caller IDs space by launching INVITE packets in such a manner that the overall entropy of these two features is not perturbed. Similarly a spam attacker can also carefully engineer spam traffic in such a way that entropy of receivers is not perturbed. However, this requires continuous monitoring of SIP traffic which is a significant overhead for an intruder and hence definitely limits his destructive capability.

The main strength of our framework lies in choice of a small number of intelligent features which monitor the spatio-temporal distribution of the traffic that makes it difficult to evade the protection mechanism using smart attacks. Mixing small floods of REGISTER and INVITE packets would certainly deteriorate the efficiency of the first temporal feature for flood attacks (Call Booking Ratio). However, if the attacker never completes the call by sending back ACKs (completing the call will cost her/him dollars that makes the attack useless), the second temporal feature (Call Establishment Ratio) will certainly raise the alarm. One might be tempted to ask why we need the Call Booking Ratio if it can be easily evaded. The answer is straightforward that a relatively high Call Booking Ratio is an early alarm of anomaly. Similarly, the spatial features such as the entropy of Call IDs and entropy of Caller IDs are measures of traffic pattern, and any perturbation in them is a sign of an attack even when some of the temporal features are compromised in a smart attack. An attacker requires greater control on the traffic flowing towards a SIP server (a daunting task) to evade both temporal and spatial features simultaneously. Therefore, our frame-

work can be a solid second line of defense for SIP servers that are operating behind the existing IP based flood detection solutions.

In the real world, end users usually get few isolated SPIT calls. The reason we focus on higher rates of SPIT traffic is that the solution is supposed to be deployed at the server and not at an end point. The spatio-temporal features that we discuss can successfully detect even small number of callers who are trying to call many users in a limited time (an advertisement call center scenario). By monitoring such a behavior, the server can label the traffic as spam (by inserting/modifying a field in the INVITE packet). The final decision is based on the policy adopted by an end user. The end user may choose to ignore the label and receive all calls, or the end user may drop all calls marked as spam unless they are from someone in the contact list. As a result, an end user who gets just one or very few spam calls can still be protected if the spam call is part of a larger spam traffic which can only be identified at the server.

Attacks on Attack Classification Module. Every classifier has a training and testing phase. During the testing phase, the classifiers can suffer from noisy training data. The only realistic attack on attack classification module is *mimicry attack* in which an attacker deliberately tries to modify traffic rates in such a way that the traffic features do not show perturbations. In order to successfully launch this type of attack, he again requires in-depth understanding of the working of our framework and its implementation.

8. Conclusion

The major contribution of this paper is a set of spatial and temporal packet based features, which detect not only different types of application layer flood attacks but also spam calls. The results of our experiments show that our features' set – using Naive Bayes classifier – is able to consistently outperform existing F-SVM scheme in all scenarios. The benefits of using packet based analysis instead of flow base are: (1) less processing overhead of feature computation module, and (2) small memory overhead of storing information. Our features are 'unsupervised' in the sense that we do not need to store their benign distributions as is done by HD. Consequently, our system can be easily deployed for realtime intrusion detection infront of a SIP server.

In order to test our PbSIP scheme with existing state-of-the-art HD and F-SVM schemes, we first collected a real-world VoIP traffic from the SIP server of 'FRIENDS'. We have developed a testbed that can inject a number of anomalies of different types in benign SIP traffic. Using our testbed, we have generated a number of threat scenarios varying in attack intensity, duration and sources. Our conclusion is that HD achieves relatively high accuracy in flood attacks but it is unable to detect spam calls. Moreover, it needs prior information of benign traffic. In contrast, the accuracy of F-SVM is low in case of harmonic attacks but it significantly improves in case of chunk attacks. However, it requires computation of a large number of flow based features. PbSIP, however, is able to show consistent accuracy on both types of attacks using only four temporal and spatial features. This shows its resilience to harmonic and chunk attacks with very low or very high attack rates – launched from single/multiple sources. We also injected spam calls into the benign traffic and PbSIP achives more than 70% accuracy on a spam rate of just 20 concurrent calls compared with 54% of F-

SVM and 0% of HD for 100% hit rate. In future, we want to enhance our system to take care of non-flooding, buffer overflow and malformed packet attacks.

Acknowledgements. This work is supported by the National ICT R&D Fund, Ministry of Information Technology, Government of Pakistan. The information, data, comments, and views detailed herein may not necessarily reflect the endorsements of views of the National ICT R&D Fund.

References

Akbar, M. 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', ACM, pp. 1419–1426.

Akbar, M., Tariq, Z. and Farooq, M. (2008), A Comparative Study of Anomaly Detection Algorithms for Detection of SIP Flooding in IMS, *in* 'IP Multimedia Subsystem Architecture and Applications, 2008 International Conference on'.

Branch, J., Giannella, C., Szymanski, B., Wolff, R. and Kargupta, H. (2012), 'In-network outlier detection in wireless sensor networks', *Knowledge and Information Systems* pp. 1–32. 10.1007/s10115-011-0474-5.
URL:`http://dx.doi.org/10.1007/s10115-011-0474-5`

Chaisamran, N., Okuda, T., Blanc, G. and Yamaguchi, S. (2011), Trust-based voip spam detection based on call duration and human relationships, *in* 'Applications and the Internet (SAINT), 2011 IEEE/IPSJ 11th International Symposium on', IEEE, pp. 451–456.

Chen, Z., Wen, W. and Yu, D. (2012), Detecting sip flooding attacks on ip multimedia subsystem (ims), *in* 'Computing, Networking and Communications (ICNC), 2012 International Conference on', IEEE, pp. 154–158.

Ehlert, S., Rebahi, Y. and Magedanz, T. (2009), 'Intrusion detection system for denial-of-service flooding attacks in sip communication networks', *International Journal of Security and Networks* **4**(3), 189–200.

Fawcett, T. (2004), 'ROC graphs: Notes and practical considerations for researchers', *Machine Learning* **31**.

Geneiatakis, D., Vrakas, N. and Lambrinoudakis, C. (2009), Performance evaluation of a flooding detection mechanism for voip networks, *in* 'Systems, Signals and Image Processing, 2009. IWSSIP 2009. 16th International Conference on', IEEE, pp. 1–5.

Gundecha, P., Barbier, G. and Liu, H. (2011), Exploiting vulnerability to secure user privacy on a social networking site, *in* 'Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 511–519.

Ipoque (2007), 'Internet Study 2007'. http://www.ipoque.com/resources/internet-studies/internet-study-2007.

Jung, T., Martin, S., Ernst, D. and Leduc, G. (2012), 'Sprt for spit: Using the sequential probability ratio test for spam in voip prevention', *Dependable Networks and Services* pp. 74–85.

Keromytis, A. (n.d.), 'A comprehensive survey of voice over ip security research', *Communications Surveys & Tutorials, IEEE* (99), 1–24.

Kumar, G., Rahul, A. and Joonuthula, K. (2011), 'Voip flood detection using jacobson fast and hellinger distance algorithms', *Journal of Communication and Computer* **8**(5), 347–353.

Liu, L. (2011), Uncovering sip vulnerabilities to dos attacks using coloured petri nets, *in* 'Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on', IEEE, pp. 29–36.

Maron, M. and Kuhns, J. (1960), 'On relevance, probabilistic indexing and information retrieval', *Journal of the Association of Computing Machinery* **7**, 216–244.

McCue, C. (2011), Operational security analytics: doing more with less, *in* 'Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 782–782.

McGann, S. and Sicker, D. (2005), An Analysis of Security Threats and Tools in SIP-Based VoIP Systems, *in* 'Second VoIP Security Workshop'.

Nassar, M., State, R. and Festor, O. (2008), Monitoring sip traffic using support vector machines, *in* 'RAID '08: Proceedings of the 11th international symposium on Recent Advances in Intrusion Detection', Springer-Verlag, Berlin, Heidelberg, pp. 311–330.

Ono, K. and Schulzrinne, H. (2009), Have i met you before?: using cross-media relations to reduce spit, *in* 'Proceedings of the 3rd International Conference on Principles, Systems and Applications of IP Telecommunications', ACM, p. 3.

Ormazabal, G., Nagpal, S., Yardeni, E. and Schulzrinne, H. (2008), 'Secure sip: A scalable prevention mechanism for dos attacks on sip based voip systems', *Principles, systems and applications of IP telecommunications. Services and security for next generation networks* pp. 107–132.

Packet vs flow-based anomaly detection (n.d.). Whitepaper, ESPHION Network Disaster Protection.

Pham, D.-S., Saha, B., Phung, D. and Venkatesh, S. (2012), 'Detection of cross-channel anomalies', *Knowledge and Information Systems* pp. 1–27. 10.1007/s10115-012-0509-6. URL:`http://dx.doi.org/10.1007/s10115-012-0509-6`

Quinlan, J. (1993), *C4.5: Programs for machine learning*, Morgan Kaufmann.

Quittek, J., Niccolini, S., Tartarelli, S. and Schlegel, R. (2006), 'Prevention of Spam over IP Telephony (SPIT)', *NEC Technical Journal* **1**(2), 114–119.

Radermacher, T. (2005), 'Spam Prevention in Voice over IP Networks', *University of Salzburg, Salzburg* .

Rafique, M., Ali Akbar, M. and Farooq, M. (2009), Evaluating dos attacks against sip-based voip systems, *in* 'Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE', IEEE, pp. 1–6.

SANS-Institute (2007), 'SANS Top-20 2007 Security Risks'. http://www.sans.org/top20/.

Sengar, H., Wang, H., Wijesekera, D. and Jajodia, S. (2006), Fast detection of denial-of-service attacks on ip telephony, *in* 'Quality of Service, 2006. IWQoS 2006. 14th IEEE International Workshop on', IEEE, pp. 199–208.

Sengar, H., Wang, H., Wijesekera, D. and Jajodia, S. (2008), 'Detecting VoIP Floods using the Hellinger Distance', *IEEE Transactions on Parallel and Distributed Systems* **19**(6), 794–805.

Sengar, H., Wang, X. and Nichols, A. (2011), Thwarting spam over internet telephony (spit) attacks on voip networks, *in* 'Quality of Service (IWQoS), 2011 IEEE 19th International Workshop on', IEEE, pp. 1–3.

Sisalem, D., Kuthan, J., Ehlert, S. and Fokus, F. (2006), 'Denial of Service Attacks Targeting a SIP VoIP Infrastructure: Attack Scenarios and Prevention Mechanisms', *IEEE Network* **20**(5), 26–31.

Tang, J., Cheng, Y. and Zhou, C. (2009), Sketch-based sip flooding detection using hellinger distance, *in* 'Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE', IEEE, pp. 1–6.

Thandeeswaran, R., Asha, A. et al. (2012), 'Novel survey on detection of ddos attack using traceback technique in voip networks', *International Journal of Mathematical Archive (IJMA)* **2**(12).

The-VoIP-Network (2008), 'VoIP Market Trends'. http://www.the-voip-network.com/voipmarket.html/.

Vaidya, J., Yu, H. and Jiang, X. (2008), 'Privacy-preserving svm classification', *Knowledge and Information Systems* **14**, 161–178. 10.1007/s10115-007-0073-7. URL:`http://dx.doi.org/10.1007/s10115-007-0073-7`

Witten, I. and Frank, E. (2005), *Data mining: Practical machine learning tools and techniques*, 2nd edn, Morgan Kaufmann.

Wu, Y., Bagchi, S., Singh, N. and Wita, R. (2009), Spam detection in voice-over-ip calls through semi-supervised clustering, *in* 'Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP International Conference on', IEEE, pp. 307–316.

Yang, B., Sato, I. and Nakagawa, H. (2011), Secure clustering in private networks, *in* 'Data Mining (ICDM), 2011 IEEE 11th International Conference on', IEEE, pp. 894–903.

Author Biographies

Muhammad Ali Akbar received his B.E. degree in Electrical Engineering from National University of Sciences and Technology (NUST), Pakistan, in 2008. He worked on VoIP security for three years at Next Generation Intelligent Networks Research Center (nexGIN RC – NUCES), Islamabad, Pakistan. He completed his M.S. in Computer Science with specialization in Computer Security from Columbia University, N.Y. in 2011. He also worked on penetration testing of a number of smartphone applications at Cigital N.Y. He is now working as an Information Security Consultant and Smartphone Development Manager at nexGIN RC, Islamabad, Pakistan. He has over four years of experience in the fields of software development, mobile application development and Penetration Testing of web and smartphone applications. He loves reviewing code for security vulnerabilities and breaking applications through penetration testing. His research interests include information & communications security, privacy and anonymity, wireless communication, machine learning, smartphone security, information assurance, and security-aware web & mobile applications design. (Web: `http://www.muhammadakbar.com`)

Muddassar Farooq received his B.E. degree in Avionics Engineering from National University of Sciences and Technology (NUST), Pakistan, in 1996. He completed his M.S. in Computer Science and Engineering from University of New South Wales (UNSW), Australia, in 1999. He completed his D.Sc. in Informatics from Technical University of Dortmund, Germany, in 2006. In 2007, he joined the NUCES, Islamabad, Pakistan, as an associate professor. Currently he is working as professor and head, department of Electrical Engineering, NUCES, Islamabad Pakistan. He is also the Director of Next Generation nexGIN RC–NUCES. He is also a winner of Presidential Award for Contribution Towards Technology. He is the author of the book "Bee-inspired Protocol Engineering: from Nature to Networks" published by Springer in 2009. He has also coauthored two book chapters in different books on swarm intelligence. He is on the editorial board of Springers Journal of Swarm Intelligence. His research interests include agent based routing protocols for fixed and mobile ad hoc networks (MANETs), nature inspired applied systems, usable security, natural computing and engineering and nature inspired computer and network security systems.

Correspondence and offprint requests to: Muhammad Ali Akbar, Next Generation Intelligent Networks Research Center (nexGIN RC), National University of Computer & Emerging Sciences (FAST-NUCES), Islamabad, Pakistan. Email: ali.akbar@nexginrc.org