A Self-Learning System for Detection of Anomalous SIP Messages

Konrad Rieck[1], Stefan Wahl[2], Pavel Laskov[1,3], Peter Domschitz[2], and Klaus-Robert Müller[1,4]

[1] Fraunhofer Institute FIRST, Intelligent Data Analysis, Berlin, Germany
[2] Alcatel-Lucent, Bell Labs Germany, Stuttgart, Germany
[3] University of Tübingen, Wilhelm-Schickard-Institute, Germany
[4] Technical University of Berlin, Dept. of Computer Science, Germany

Abstract. Current Voice-over-IP infrastructures lack defenses against unexpected network threats, such as zero-day exploits and computer worms. The possibility of such threats originates from the ongoing convergence of telecommunication and IP network infrastructures. As a countermeasure, we propose a self-learning system for detection of unknown and novel attacks in the Session Initiation Protocol (SIP). The system identifies anomalous content by embedding SIP messages to a feature space and determining deviation from a model of normality. The system adapts to network changes by automatically retraining itself while being hardened against targeted manipulations. Experiments conducted with realistic SIP traffic demonstrate the high detection performance of the proposed system at low false-positive rates.

1 Introduction

Voice-over-IP (VoIP) infrastructures provide a replacement of current circuit-switched networks. VoIP and IP multimedia subsystem (IMS) technology reduces deployment costs and provides extensive functionality to operators and end users. The advent of VoIP and IMS technology, however, gives rise to new security threats originating from network-based as well as service-based vulnerabilities. For instance, IP networks connected to the Internet are plagued by network attacks and malicious software. Unfortunately, VoIP infrastructures inherently possess properties attractive to developers of malicious software:

1. *Diversity.* Enterprise VoIP infrastructures consist of a large amount of heterogeneous network nodes covering mobile and wired end devices as well as gateway and registration servers of various manufacturers and brands. A single security breach in any of these nodes suffices to infiltrate the infrastructure, e.g, to eavesdrop communication at compromised nodes.
2. *Availability.* A second inherent property of VoIP infrastructures is availability, which is necessary for unimpeded communication between network nodes. Malicious software, such as a potential "VoIP worm", might exploit this property to rapidly propagate through the infrastructure comprising the vast majority of vulnerable nodes in a matter of minutes [30].

3. *Lack of transparency.* Terminal devices of VoIP services usually hide network and operating system details such as running processes and services from the end user. For instance, given a smartphone with VoIP capabilities it is hard to assess, whether the system has been compromised. Malicious software not disrupting functionality may control VoIP devices for a long period of time without being detected, for example to distribute unsolicited content.

It is likely from these features that current security threats will enter the realm of VoIP infrastructures in the near future. Especially the increasing commercialization of malicious software may further advance this development, e.g., as observed for the computer worm "Storm" [8].

A large body of research has focused on security defenses specific to IP telephony, such as the identification of fraudulent usage, the detection of denial-of-service attacks and the recognition of unsolicited content. Various concepts of misuse detection have been studied in the field of VoIP security, e.g., intrusion detection systems based on signatures [6, 19], rules [4, 36], protocol specifications [28, 32] and VoIP honeypots [15, 17]. Yet few research has considered the detection of *unknown and novel network attacks*, which arise with the appearance of zero-day exploits and computer worms. Systems based on misuse detection do not address this problem, as signatures or rules need to be available prior to the emerging security threats.

In this contribution, we propose a *self-learning system* for detection of unknown and novel attacks in the Session Initiation Protocol (SIP), which complements current VoIP security measures. The system enables identification of anomalous content by embedding SIP messages to a feature space and determining deviation from a model of normality. The system is "self-learning", as it is capable to automatically retrain itself in order to adapt to moderate changes in the network environment and traffic. Moreover, the retraining process is hardened against targeted manipulation. Experiments conducted on realistic SIP traffic and anomalous messages generated using a security testing tool demonstrate the high effectiveness of the proposed system at low false-positive rates – a criterion essential for practical deployment.

The rest of this paper is structured as follows: The self-learning system is introduced in Section 2 covering details on feature extraction, anomaly detection and retraining. Experiments on detection and run-time performance of the system with SIP traffic are presented in Section 3. Related work on VoIP intrusion detection is discussed in Section 4 and the paper is concluded in Section 5.

2 A Self-Learning System

To protect VoIP infrastructures from unknown network attacks, we introduce a *self-learning system* for anomaly detection in the Session Initiation Protocol (SIP [26]). SIP is a widely used protocol for signaling communication and transmission of media in VoIP and IMS infrastructures. Network attacks targeting a VoIP system may occur in any element or content of incoming SIP messages; hence, our self-learning system is designed to analyze complete SIP messages as raw byte sequences – eliminating the need of preprocessing and normalization procedures. The architecture of our system is illustrated in Figure 1.

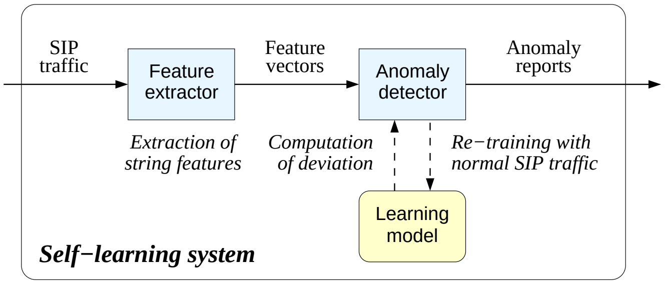


Fig. 1. Architecture of the self-learning system for SIP anomaly detection.

The basic processing stages of the system during operation are outlined in the following and discussed in further detail in the rest of this section.

1. *Feature extraction.* Incoming SIP messages are analyzed using a set of feature strings. Based on the occurrences of these strings, each messages is mapped to a feature vector reflecting individual characteristics of the message as captured by the feature strings. This feature extraction is covered in Section 2.1.
2. *Anomaly detection.* The feature vectors corresponding to SIP messages are compared against a model of normality. This model is either detecting global or local anomalies by computing distances in the underlying vector space. Anomalous SIP messages are flagged and reported by the system. The detection process is described in Section 2.2.
3. *Initialization & retraining.* On initial deployment of the system as well as on a periodic basis the learning model is updated using traffic flagged as normal. To prevent external manipulation of the learning process randomization, sanitization and verification of the model are performed. The initialization and retraining process is discussed in Section 2.3.

2.1 Feature Extraction

The syntax and structure of SIP messages is defined by the SIP protocol specification [26], yet such structure is not suitable for application of anomaly detection methods, as these usually operate on vectorial data. To address this issue we derive a technique for embedding SIP messages to a high-dimensional vector space, which reflects typical characteristics of the observed SIP traffic. This embedding has been successfully applied in the context of network intrusion detection [23] and its efficient implementation is detailed in [24].

A SIP message corresponds to a sequence of bytes and its content can be characterized by frequencies of contained substrings. For instance, the substrings

"`From`", "`To`" and "`Via`" play an important role in the semantics of SIP. We define a set of *feature strings* S to model the content of SIP messages. Given a feature string $s \in S$ and a SIP message x, we determine the number of occurrences of s in x and obtain a frequency value $f(x, s)$. The frequency of s acts as a measure of its importance in x, e.g., $f(x, s) = 0$ corresponds to no importance, while $f(x, s) > 0$ reflects the contribution of s in x.

An embedding function ϕ maps all SIP messages X to an $|S|$-dimensional vector space by considering the frequencies of feature strings in S:

$$\phi : X \to \mathbb{R}^{|S|} \quad \text{with} \quad \phi(x) \mapsto \big(f(x, s)\big)_{s \in S}$$

For example, if S contains the strings "`foo.org`" and "`john`", two dimensions in the resulting vector space correspond to the frequencies of these strings in SIP messages. Hence, the communication of a user "`john`" with a network node in the domain "`foo.org`" would be reflected in high frequencies of these strings in the respective SIP traffic.

However, it is impractical to define a set of feature strings S a priori, simply because not all important strings are known in advance, e.g., the user "`john`" might not be registered with the VoIP infrastructure when the self-learning system is deployed. To solve this problem the set of feature strings S is defined *implicitly* by introduction the notion of *tokens* and *n-grams*.

Tokens. SIP is a text-based protocol, thus, its content can be described in terms of textual tokens and words. An implicit set of feature strings in this view corresponds to all possible strings separated by specific delimiter symbols. If we denote all byte values by B and define $D \subset B$ as delimiter symbols, a set S referred to as *tokens* is given by

$$S := (B \setminus D)^*,$$

where $*$ is the Kleene closure corresponding to all possible concatenations of a set. The resulting set S has an infinite size, since strings of any length containing bytes from $(B \setminus D)$ are contained in S. A SIP message, however, comprises only a limited amount of such strings as the number of partitioned substrings in a message is bounded by its length.

The following example illustrates how a simplified SIP message is mapped to a vector space using the notion of tokens, where the set of delimiters is $D = \{\square, \texttt{@}, \texttt{:}, \texttt{/}\}$.

$$\phi(\ \texttt{BYE}\square\texttt{SIP:JOHN@DOE}\square\texttt{SIP/2.0}\ ) \ \longmapsto \ \begin{array}{cc} \left(\begin{matrix}1\\2\\1\\1\\1\end{matrix}\right) & \begin{matrix}\texttt{BYE}\\\texttt{SIP}\\\texttt{JOHN}\\\texttt{DOE}\\\texttt{2.0}\end{matrix}\end{array}$$

The vector at the right comprises frequency values for each token in the simplified SIP message. For instance, the two occurrences of the token "`SIP`" are reflected in the second column of the feature vector.

The granularity of feature extraction based on tokens can be controlled using the delimiter set D. The less delimiters are defined the more specific are the extracted tokens. For our self-learning system, we define the following delimiter symbols capturing generic SIP tokens such as header names, header values, recipients and attribute strings.

$$D = \{\,\texttt{:}, \texttt{;}, \texttt{,}, \texttt{=}, \texttt{<}, \texttt{>}, \texttt{/}, \texttt{SPC}, \texttt{CR}, \texttt{LF}\}$$

N-grams. Tokens are intuitive and expressive to the human analyst, still they may not always identify anomalous content of novel attacks, due to the definition of delimiter symbols in advance. An alternative technique for implicit definition of feature strings S are so called n-grams. Instead of partitioning a SIP message, feature strings are extracted by moving a sliding window of length n over the message content. At each position a substring of length n is considered and its occurrences are counted. Formally, the set of feature strings S referred to as *n-grams* is defined as

$$S := B^n,$$

where B^n corresponds to all possible strings of length n from the set B.

For example, if $n = 4$ we obtain 4-grams, which for the simplified SIP message considered in the previous section yields the following embedding to a feature vector space.

$$\phi(\;\texttt{BYE}\square\texttt{SIP:JOHN@DOE}\square\texttt{SIP/2.0}\;) \;\longmapsto\; \begin{array}{cc} \begin{pmatrix} 1 \\ 1 \\ 2 \\ 2 \\ \vdots \end{pmatrix} & \begin{array}{l} \texttt{BYE}\square \\ \texttt{YE}\square\texttt{S} \\ \texttt{E}\square\texttt{SI} \\ \square\texttt{SIP} \\ \quad\vdots \end{array} \end{array}$$

Note, that similar to the previous example, the occurrences of the term “SIP” are reflected in the feature vector. In particular, the 4-gram “E□SI” is contained twice and its frequency is given in the third column of the feature vector. To simplify presentation further 4-grams are not shown in the example.

The vector space induced by n-grams is high-dimensional, e.g., for $n = 4$ there exist 256^4 different dimensions. Moreover, in the case of tokens the resulting space has infinite dimension as the underlying set S has infinite size. Computing and comparing vectors in such high-dimensional spaces seems infeasible at a first glance. However, for both types of features – n-grams and tokens – the number of feature strings contained in a single SIP message is linear in its length.

As a consequence, a SIP messages x of length l comprises at most l different n-grams or tokens, that is $\mathcal{O}(l)$ dimensions are non-zero in $\phi(x)$. This sparsity of the embedding $\phi(x)$ can be exploited to derive linear-time methods for extraction and comparison of feature vectors [24], which ultimately enables efficient anomaly detection over embedded SIP messages.

2.2 Anomaly Detection

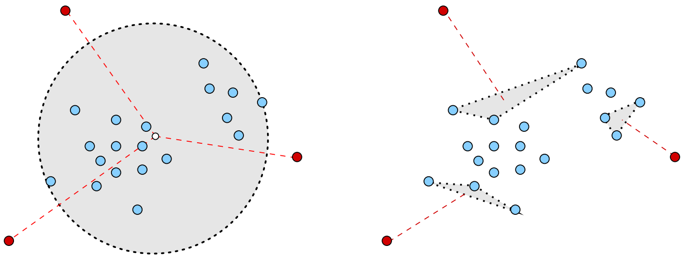
An important extension to current VoIP security is the detection of unknown network attacks emerging from IP networks. Anomaly detection addresses this problem and complements signature-based analysis by modeling profiles for "normality". Although anomaly detection methods have been successfully applied in various incarnations of network intrusion detection, such as for identification of anomalous packet headers [13, 14] or payloads [23, 34], all methods share the same concept – *anomalies are deviations from a learned model of normality* – and only differ in concrete notions of normality and deviation.

The embedding of SIP messages to a vector space introduced in Section 2.1 enables expressing normality and deviation *geometrically*, which yields intuitive yet powerful learning models for anomaly detection. The basis for such geometric learning models is a distance function d, which assess the dissimilarity of two messages x and z by

$$d(x, z) = ||\phi(x) - \phi(z)||$$

and corresponds to a Euclidean distance in the vector space. Messages originating from a similar context, such as consecutive telephone calls, yield low distances and lie close to each other, while messages from different contexts, such as calls monitored at distinct locations, result in higher distances and are separated from each other. In this geometric view SIP messages are associated with points forming groups and clouds in the induced vector space depending on the underlying semantics and context.

For our self-learning system we focus on two simple realizations of geometric anomaly detection, which build on a *global* and *local* concept of normality and deviation thereof. Before introducing these concepts, we need to establish some notation. We denote the set of SIP messages used for learning by $X = \{x_1, \ldots, x_n\}$ and refer to a new incoming message as z. During the learning and anomaly detection process all messages $x_i \in X$ and z are represented as vectors $\phi(x_i)$ and $\phi(z)$ using the embedding function ϕ introduced in Section 2.1.



(a) Global

(b) Local

Fig. 2. Geometric anomaly detection.

Global Detection. Network attacks often significantly deviate from normal traffic in terms of contained substrings. Thus, it is natural to define anomaly detection using a *global* model of normality capturing properties shared by the majority of X. A simple geometric shape reflecting this concept is a hypersphere. Normality is modeled by placing a hypersphere around the vectors of X and deviation is determined by the distance from the center μ of the hypersphere. Figure 2(a) illustrates a hypersphere enclosing a set of points, where anomalies are identified by large distances from the center.

The *smallest enclosing hypersphere* – the optimal model of normality – can be determined by solving the following optimization problem

$$\mu^* = \operatorname*{argmin}_{\mu} \max_{1 \leq i \leq n} ||\phi(x_i) - \mu||, \tag{1}$$

which returns the center μ^* of the hypersphere with the smallest radius containing all points in X. Unfortunately, unknown attacks in X may spoil this process and lead to hyperspheres with larger volume. This problem is alleviated by the technique of *regularization*, which "softens" the margin of the hypersphere, such that outliers and unknown attacks can be compensated. An introduction to this regularized learning model is provided in [12, 31], covering the respective theory as well as the efficient computation implemented in our self-learning system.

Once the center μ^* has been found, deviation δ from the model of normality is determined by computing the distance of an incoming message z from μ^*,

$$\delta(z) = ||\phi(z) - \mu^*||. \tag{2}$$

Application of the learned model in Equation (2) requires computing only a single distance value for each incoming message, as μ^* is fully determined from X during the prior learning phase.

Local Detection. If the SIP traffic monitored at a network node is inherently heterogeneous, e.g., at a large gateway, a global model of normality might not suffice for detection of unknown and novel attacks. The embedded messages are geometrically distributed in different clusters of points hindering application of a single enclosing hypersphere. To address this issue we extend our self-learning system with a *local* anomaly detection scheme, which assesses deviation of a message by considering only a fraction of messages in the training data.

A local model of normality can be derived using the notion of k-nearest neighbors. We define the neighbors of a vector $\phi(z)$ using a permutation π of X, such that the embedded message $x_{\pi[i]}$ is the i-th nearest neighbor of z in terms of distances. In other words, π sorts the vectors in X according to their distance from z in ascending order. A simple deviation δ_s from this model is calculated as the average distance of $\phi(z)$ to its k-nearest neighbors and given by

$$\delta_s(z) = \frac{1}{k} \sum_{i=1}^{k} ||\phi(z) - \phi(x_{\pi[i]})||. \tag{3}$$

Messages strongly deviating from their k-nearest neighbors yield a large average distance, while messages close to their neighbors get a low deviation score. Figure 2(b) illustrates the concept of k-nearest neighbors for $k = 3$. Anomalies deviate in Figure 2(b) from local normality in that they show a large average distance to the respective three neighboring points.

The average distance to a set of neighbors, however, is density-dependent. Points in dense regions yield low deviations, while points in sparse areas are flagged as anomalous, although they do not constitute attacks. Thus, we refine the deviation δ using the average distance between the k-nearest neighbors as normalization term

$$\begin{aligned}\delta(z) =& \frac{1}{k}\sum_{i=1}^{k} ||\phi(z) - \phi(x_{\pi[i]})|| \\ &- \frac{1}{k^2}\sum_{i=1}^{k}\sum_{j=1}^{k} ||\phi(x_{\pi[j]}) - \phi(x_{\pi[i]})||.\end{aligned} \tag{4}$$

The first term emphasizes points that lie far away from its neighbors, whereas the second term discounts abnormality of points in wide neighborhood regions.

In contrast to the global model in Equation (2), computing Equation (4) requires determining several distance values. In particular, for each incoming message $\mathcal{O}(|X|k^2)$ distance computations need to be performed for finding the k-nearest neighbors and calculating δ. Hence, for the local model of normality the amount of learning data X need to be constrained to achieve effective run-time performance. Experiments on the run-time as well as detection performance of the global and local anomaly detection methods are presented in Section 3.

2.3 Initialization and Retraining

Retraining enables the self-learning system to adapt itself to changes in the network environment, such as the presence of new terminal nodes or media services. To achieve this goal the learning model is trained on a periodic basis using network traffic previously flagged as normal. The interval of these retraining cycles depends on the monitored volume of SIP traffic and the estimated rate of changes in the network environment. For instance, devices processing millions of SIP messages per day might demand updates on a daily basis, while minor network nodes are sufficiently adapted in weekly or even monthly intervals.

For the initial deployment of the self-learning system, we assume that a coarse model of normality is already available, e.g., from another running system or generated using prototypical SIP traffic for the particular VoIP infrastructure. We thus restrict our scope to the retraining procedure, as initialization basicly resembles this process.

While automatic retraining provides ease of use to an operator, it introduces a new security vulnerability: attacks and anomalies in the training data may tamper learning and impede attack detection. In particular, an adversary could attempt to "poison" the learning model during retraining using specifically crafted SIP messages, such that later attacks targeted against the system are not detected [9]. Thus, defenses against targeted manipulation of our learning system need to be provided.

Manipulation Defense. As a first defense against manipulations and unknown attacks the running self-learning system is applied to any potential training data, eliminating all attacks detectable using the present model of normality. To further harden the system against adversarial manipulation the following defense techniques are considered.

(a) *Randomization.* The traffic volume in enterprise VoIP infrastructures is huge and due to storage constraints only a limited fraction can be used for retraining. Instead of choosing a fixed partition, the self-learning system is retrained with randomly drawn samples which are collected from the monitored traffic between update cycles.

(b) *Sanitization.* The collected data is passed to a sanitization procedure filtering out irregular events, e.g., as proposed for network intrusion detection in [3]. In our self-learning system the collected SIP messages are sorted according to their deviation score and messages yielding the highest deviations are removed, e.g. a fraction of 5%-10%.

(c) *Verification.* Once a new model is trained it is applied concurrently with the previous one. As the new model originates from recent traffic, it is supposed to report similar or lower average deviation in comparison to the old. If after a fixed verification period the observed average deviation of the new model is too high, the update process fails and the model is discarded.

These defense methods particularly harden targeted manipulations against the self-learning system. On the one hand, randomization forces an attacker to constantly provide manipulated SIP messages to the system in order to resolve the random sampling. On the other hand, if the attacker sends too many manipulated messages to the system, the retrained model of normality will significantly deviate from normal traffic and, thus, a comparison with the old learning model will indicate various false anomalies. Finally, if an attacker controls the majority of traffic, he can be identified using techniques for detection of denial-of-service attacks, as proposed for instance in [21, 27, 29].

Calibration. As a last issue related to initialization and retraining of our self-learning system, we present a calibration procedure, which automatically provides a threshold for anomaly detection discriminating legitimate SIP traffic from anomalous or attack messages.

The calibration procedure builds on the concept of *cross-validation.* The preprocessed and sanitized training data is segmented into k partitions of equal size. A learning model is then trained on the SIP messages of $k-1$ partitions and applied on the l messages of the remaining partition for computation of deviations scores $D = \{\delta_1, \ldots, \delta_l\}$. This process is repeated k times, such that for each partition i individual deviation scores D_i are determined.

A threshold t is then computed using the largest deviation scores in each partition D_i as defined by

$$t = \frac{1}{k} \sum_{i=1}^{k} \max(D_i), \quad v = \frac{1}{k} \sum_{i=1}^{k} (\max(D_i) - t)^2,$$

where t corresponds to the mean of the largest deviation scores and v to the empirical variance. The rational underlying this definition of t is that outliers and unknown attacks have been filtered from the training data and thus the largest deviation scores correspond to unusual but still legitimate traffic. The threshold is determined as the average of these scores, such that similar traffic is still accepted by the system.

The variance v acts as criterion for assessing the quality of the generated threshold. The randomization discussed in the previous section provides uniformly distributed samples from the running traffic, so that a high empirical variance of the threshold indicates irregularities in the training data. In this case the retraining process is aborted and the current calibrated learning model remains in operation.

3 Experiments

In the previous sections we have introduced the concept of a self-learning system for anomaly detection. To assess the capabilities of such a system in practice we conducted experiments on SIP traffic and artificial attacks generated using a security testing tool. In particular, we were interested to (a) evaluate the detection performance of our self-learning system on unseen attacks and (b) provide results for the run-time performance on real SIP traffic.

3.1 Evaluation Data

For our experiments we generated an evaluation data set comprising SIP request and response messages. These SIP traces contain contiguous SIP dialogs from a single SIP terminal as well as interleaved SIP dialogs recorded at a network edge ingress where multiple terminals are connected to. The messages originated mainly from several NGN test labs where multiple services and interworking tests are performed. Others are derived from research demonstrator setups where new services and functions are elaborated. The final portion of SIP traces are anonymized original signaling messages. This composition guarantees a very broad spectrum of correct SIP messages which partly contain Session Description Protocol (SDP [7]) payloads.

In contrast to Internet services, only few network attacks against SIP-based devices have been disclosed, most notably attacks identified using fuzzing techniques [1]. In the absence of a large collection of SIP attacks, we conducted our experiments using artificially generated attacks. A VoIP version of the security and syntax testing tool *Codenomicon Defensics*[5] was applied to produce several thousand anomalous SIP messages – covering syntactical anomalies as well as security probes for boundary condition, format string and input validation vulnerabilities. The generated anomalous SIP messages are post-processed to eliminate any remaining redundancy by permuting the sequence of the header fields and randomizing certain header and parameter values. This post-processing takes care that the original anomalous properties of each message persist, while detection via artifacts specific to the testing tool is largely prevented.

The resulting evaluation data set contains 4428 normal and 9999 anomalous SIP messages as raw byte sequences, where headers from the network and transport layer have been removed from each message. For all experiments the data set is split into *disjunct* training and testing partitions. The training data was used for learning models of normality and determining optimal model parameters, while the testing data was applied for generating results using the trained learning models.

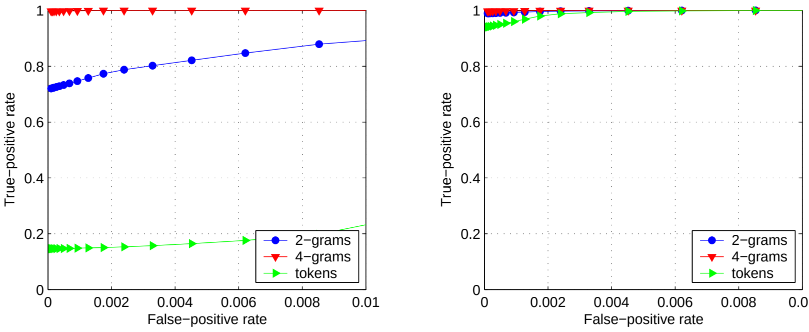
3.2 Detection Performance

In order to evaluate the detection performance of our self-learning system, we implemented the feature extraction and anomaly detection methods presented in Section 2.1 and Section 2.2 in a prototypical system. Using this system we performed the following experimental procedure: 1,000 normal SIP messages are drawn from the training data, and models of normality are learned for different model parameters, such as the neighborhood size of the local anomaly detection method. The learning model achieving the best accuracy on the training data is then evaluated on 500 normal and 500 anomalous messages randomly drawn from the testing data. The procedure is repeated over 50 runs and the results are averaged.

Figure 3 depicts the detection performance of the self-learning system using 2-grams, 4-grams and tokens as string features. Figure 3(a) shows results for the global anomaly detection method and Figure 3(b) for the local anomaly detection method. The performance is presented as receiver operating characteristic, in short *ROC*, curves which show the false-positive rate of a methods on the x-axis and the true-positive rate on the y-axis for different thresholds. High detection accuracy is reflected in the top left of a ROC curve, while random detection corresponds to a diagonal line. Note that in Figure 3 the true-positive rate is given in the full interval 0 to 100%, while the false-positive rate shows only the range from 0 to 1%.

The local anomaly detection method yields a significantly higher detection accuracy in comparison to the global detection method. In particular, for all types of feature strings a true-positive rate over 97% is achieved *with no false-positives*. Moreover, for the 4-grams features over 99% of the attacks are detected – even though all attacks were unknown to the system during application. For

[5] Codenomicon DEFENSICS, `http://www.codenomicon.com`



(a) Global anomaly detection (b) Local anomaly detection

Fig. 3. Detection performance of the self-learning system for different string features.

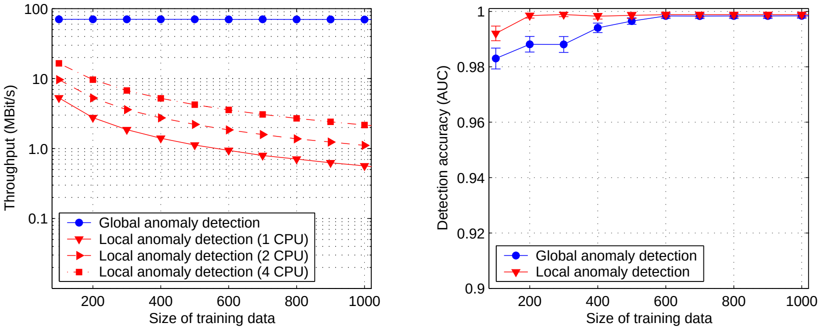
the global anomaly detection method only the 4-gram features enable similar accuracy and in contrast the token features provide a very poor detection performance.

As the evaluation data used in our experiments originates from different sources, it expresses heterogeneous characteristics particularly suitable for application of a local anomaly detection method. The superior results presented in Figure 3(b) confirm this finding on real SIP traffic. Furthermore, the embedding to a vector space using 4-grams enables a very effective discrimination of normal traffic and attacks – by capturing particular substrings related to normal or anomalous messages – so that even the global method yields a high detection accuracy in the underlying vector space.

3.3 Run-time Performance

In practice, the effectiveness of an intrusion detection system is determined by the detection rate as well as the run-time performance. In order to analyze the run-time of our self-learning system, we conducted experiments on a standard server system using AMD Opteron CPUs. The run-time, however, strongly depends on the complexity of the applied model of normality, which in turn is learned from provided training data. To model this effect, we varied the number of SIP messages used for learning from 100 to 1000 messages. For each size, we monitored the run-time performance of the system as well as the achieved detection accuracy.

In particular, we performed the following experimental procedure: 1,000 normal SIP messages are drawn from the evaluation data set and the run-time is measured for feature extraction and anomaly detection using a previously trained learning model. Based on the length of each SIP message the throughput of the system is estimated in terms of Megabits per second. The detection accuracy is determined using the setup applied in the previous experiment. The procedure is repeated 50 times and the results are averaged. As string features we focus on 4-grams, which yield the best detection in the previous experiment.



(a) Run-time performance

(b) Detection accuracy

Fig. 4. Run-time and detection performance for different sizes of training data.

Figure 4 details the results of this experiment, where Figure 4(a) shows the run-time performance for varying size of training data and Figure 4(b) shows the corresponding detection accuracy of the self-learning system. The detection accuracy is given as *area under the ROC curve* (AUC), which simply integrates the true-positive rate over the false-positive rate of a ROC curve.

The run-time performance of the global anomaly detection method does not depend on the size of the training data. As discussed in Section 2.2 the training data is used to determine a single vector μ^* as learning model, and thus, the run-time is independent of the training size. The accuracy of the global anomaly detection scheme reaches an AUC value near 99% at around 600 training examples. On average a total of 70 Mbit per second in terms of SIP messages are processed using the global anomaly detection scheme.

In comparison the local anomaly detection method significantly depends on the size of the training data, as for each incoming data point a set of corresponding nearest neighbors need to be determined and evaluated. This process, however, can be easily parallelized, so that Figure 4(a) reports the run-time for a single CPU as well as SMP implementations with 2 and 4 CPUs. In this experiment, a detection accuracy at 99% is reached using a training data size of 300 instances which corresponds to a throughput between 2 and 6 Mbit per second depending on the number of CPUs.

4 Related Work

Security has been an active area of research in the domains of VoIP and IMS technology. Specifically for the SIP protocol considerable effort has been spent on identification and categorization of security threats [5, 18, 33]. Among these threats, attacks targeting the availability of VoIP play a salient role and much research has studied specific methods for detection and mitigation of denial-of-service attacks [21, 27, 29, 37].

Beside specific solutions, various concepts for generic VoIP intrusion detection have been proposed in the community. Sengar et al. [28] and Truong et al. [32] devise specification-based detection systems using finite state machines for modeling VoIP protocols. Moreover, rule-based detection frameworks have been proposed – namely Scidive [36] and VoIP defender [4] – which provide efficient identification of VoIP attacks at different and across protocol layers.

Attack signatures as commonly used in network intrusion detection systems such as Snort [25] or Bro [20] were expanded to the SIP protocol by Niccolini et al. [19] and Apte et al. [2]. Similarly, Geneiatakis et al. [6] derived specific signatures for detection of malformed message content in the SIP protocol.

A different approach for detection of VoIP attacks using honeypots was proposed by Nassar et al. [15, 17], covering an individual "Honeyphone" as well as a network of emulated SIP devices. In both scenarios, attacks are identified when contacting the fake devices and detection rules are derived using Bayesian inference [16].

The self-learning system proposed in this contribution differs from previous research in VoIP security, as it does not require providing detection rules, signatures or protocol specifications prior to deployment. In particular, the system exploits characteristics of normal SIP traffic monitored at the network and, hence, enables identification of yet unknown attacks for which no signatures are available. In this view, it is similar to anomaly detection concepts proposed by Kruegel et al. [10, 11], Wang et al. [34, 35] and Rieck et al. [22, 23] for network intrusion detection.

5 Conclusions

Modern telecommunication in form of VoIP and IMS infrastructures requires effective defense against sudden network attacks. Current techniques for misuse detection such as signature-based intrusion detection systems fail to cope with fast emerging threats as appropriate attack signatures need to be available prior to a security incident.

We address this problem by introducing a self-learning system for detection of unknown and novel attacks in the SIP protocol. Our system proceeds by embedding SIP messages to a high-dimensional vector space defined over substrings contained in the messages. The vectorial representation induces geometric relations between SIP messages and enables the formulation of global and local anomaly detection methods. Using these methods the self-learning system generates a model of normality from given SIP traffic and identifies anomalous content in incoming SIP messages. Furthermore, the system supports retraining the model of normality automatically to adapt itself to changes in the network environment. The retraining process is hardened against manipulation and unknown attacks in the training data using randomization and sanitization techniques.

Experiments conducted on realistic SIP traffic and anomalous messages generated using a security testing tool demonstrate the high effectiveness of the proposed system. In particular, a prototypical implementation achieved a detection rate of over 99% with no false-positives in the corresponding experiments. Depending on the applied anomaly detection method, the system is able to process SIP traffic up to 70 Megabits per second, while still providing high accuracy.

Although the realized throughput of our implementation does not yet comply with recent products for the VoIP and IMS market targeting Gigabit networks, it may provide defense in combination with filtering techniques relieving the impact of high traffic volumes. Future research will focus on techniques for pre-filtering of SIP traffic as well as development of run-time improvements.

Bibliography

[1] H. Abdelnur, O. Festor, and R. State. KiF: A statefule SIP fuzzer. In *Proc. of International Conference on Principles, Systems and Applications of IP Telecommunications (IPTCOMM)*, pages 47–56, 2007.

[2] V. Apte, Y.-S. Wu, S. Garg, and N. Singh. SPACEDIVE: A distributed intrusion detection system for voice-over-ip environments. In *Abstract Paper at International Conference on Dependable Systems and Networks (DSN)*, 2006.

[3] G. Cretu, A. Stavrou, M. Locasto, S. Stolfo, and A. Keromytis. Casting out demons: Sanitizing training data for anomaly sensors. In *ieeesp*, 2008. to appear.

[4] J. Fiedler, T. Kupka, S. Ehlert, T. Magedanz, and D. Sisalem. VoIP Defender: Highly scalable SIP-based security architecture. In *Proc. of International Conference on Principles, Systems and Applications of IP Telecommunications (IPTCOMM)*, pages 11–17, 2007.

[5] D. Geneiatakis, T. Dagiuklas, G. Kambourakis, C. Lambrinoudakis, S. Gritzalis, S. Ehlert, and D. Sisalem. Survery of security vulnerabilities in session initial protocol. *IEEE Communications Surverys & Tutorials*, 8(3):68–81, 2006.

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

[7] M. Handley, V. Jacobson, and C. Perkins. SDP: Session Description Protocol. RFC 4566 (Proposed Standard), July 2006.

[8] T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. Freiling. Measurements and mitigation of peer-to-peer-based botnets: A case study on storm worm. In *First USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2008.

[9] M. Kloft and P. Laskov. A poisoning attack against online anomaly detection. In *NIPS Workshop on Machine Learning in Adversarial Environments for Computer Security*, 2007.

[10] C. Kruegel, T. Toth, and E. Kirda. Service specific anomaly detection for network intrusion detection. In *Proc. of ACM Symposium on Applied Computing*, pages 201–208, 2002.

[11] C. Kruegel and G. Vigna. Anomaly detection of web-based attacks. In *Proc. of 10th ACM Conf. on Computer and Communications Security*, pages 251–261, 2003.

[12] P. Laskov, C. Gehl, S. Krüger, and K. R. Müller. Incremental support vector learning: Analysis, implementation and applications. *Journal of Machine Learning Research*, 7:1909–1936, Sept. 2006.

[13] W. Lee, S. Stolfo, and K. Mok. A data mining framework for building intrusion detection models. In *Proc. of IEEE Symposium on Security and Privacy*, pages 120–132, 1999.

[14] M. Mahoney. Network traffic anomaly detection based on packet bytes. In *Proc. of ACM Symposium on Applied Computing*, pages 346 – 350, 2003.

[15] M. Nassar, S. Niccolini, R. State, and T. Ewald. Holistic VoIP intrusion detection and prevention system. In *Proc. of International Conference on Principles, Systems and Applications of IP Telecommunications (IPTCOMM)*, pages 1–9, 2007.

[16] M. Nassar, R. State, and O. Festor. Intrusion detection mechanisms for VoIP applications. In *Proc. of VoIP Security Workshop (VSW)*, 2006.

[17] M. Nassar, R. State, and O. Festor. VoIP honeypot architecture. In *Proc. of IEEE Symposium on Integrated Network Management (IM)*, pages 109–118, 2007.

[18] S. Niccolini. VoIP security threats. Draft of IETF Working Group Session Peering for Multimedia Interconnect (SPEERMINT), 2006.

[19] S. Niccolini, R. Garroppo, S. Giordano, G. Risi, and S. Ventura. SIP intrusion detection and prevention: recommendations and prototype implementation. In *Proc. of IEEE Workshop on VoIP Management and Security*, pages 47–52, 2006.

[20] V. Paxson. The bro 0.8 user manual. Lawrence Berkeley National Laboratroy and ICSI Center for Internet Research, 2004.

[21] B. Reynolds and D. Ghosal. Secure IP telephony using multi-layered protection. In *Proc. of Network and Distributed System Security Symposium (NDSS)*, 2003.

[22] K. Rieck and P. Laskov. Detecting unknown network attacks using language models. In *Detection of Intrusions and Malware, and Vulnerability Assessment, Proc. of 3rd DIMVA Conference*, LNCS, pages 74–90, July 2006.

[23] K. Rieck and P. Laskov. Language models for detection of unknown attacks in network traffic. *Journal in Computer Virology*, 2(4):243–256, 2007.

[24] K. Rieck and P. Laskov. Linear-time computation of similarity measures for sequential data. *Journal of Machine Learning Research*, 9(Jan):23–48, 2008.

[25] M. Roesch. Snort: Lightweight intrusion detection for networks. In *Proc. of USENIX Large Installation System Administration Conference LISA*, pages 229–238, 1999.

[26] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard), June 2002. Updated by RFCs 3265, 3853, 4320, 4916.

[27] H. Sengar, H. Wang, D. Wijesekera, and S. Jajodia. Fast detection of denial of service attacks on ip telephony. In *Proc. of International Workshop on Quality of Service (IWQoS)*, pages 199–208, 2006.

[28] H. Sengar, D. Wijesekera, H. Wang, and S. Jajodia. VoIP intrusion detection through interacting protocol state machines. In *Proc. of International Conference on Dependable Systems and Networks (DSN)*, pages 393 – 402, 2004.

[29] D. Sisalem, J. Kuthan, and S. Ehlert. Denial of service attacks targeting a SIP VoIP infrastructure: Attack scenarios and prevention mechanisms. *IEEE Networks Magazine*, 20(5), 2006.

[30] S. Staniford, V. Paxson, and N. Weaver. How to 0wn the internet in your spare time. In *Proc. of USENIX Security Symposium*, 2002.

[31] D. Tax and R. Duin. Support vector domain description. *Pattern Recognition Letters*, 20(11–13):1191–1199, 1999.

[32] P. Truong, D. Nieh, and M. Moh. Specification-based intrusion detection for H.232-based voice over IP. In *Proc. of IEEE Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 387–392, 2005.

[33] VoIPSA. Voip security and privacy threat taxonomy. Report of Voice over IP Security Alliance, 2005.

[34] K. Wang, J. Parekh, and S. Stolfo. Anagram: A content anomaly detector resistant to mimicry attack. In *Recent Adances in Intrusion Detection (RAID)*, pages 226–248, 2006.

[35] K. Wang and S. Stolfo. Anomalous payload-based network intrusion detection. In *Recent Adances in Intrusion Detection (RAID)*, pages 203–222, 2004.

[36] Y.-S. Wu, S. Bagchi, S. Garg, and N. Singh. SCIDIVE: a stateful and cross protocol intrusion detection architecture for voice-over-ip environments. In *Proc. of International Confernce on Dependable Systems and Neteworks (DSN)*, pages 433–442, 2004.

[37] G. Zhang, S. Ehlert, T. Magedanz, and D. Sisalem. Denial of service attack and prevention on SIP VoIP infrastructures using DNS flooding. In *Proc. of International Conference on Principles, Systems and Applications of IP Telecommunications (IPTCOMM)*, 2007.