Risk Management in VoIP Infrastructures using Support Vector Machines

Mohamed Nassar, Oussema Dabbebi, Rémi Badonnel, Olivier Festor

Risk Management in VoIP Infrastructures using Support Vector Machines

M. Nassar, O. Dabbebi, R. Badonnel, O. Festor
INRIA Grand Est - LORIA Research Center
54602 Villers-lès-Nancy, France
{nassar, dabbebi, badonnel, festor}@loria.fr

***Abstract*—Telephony over IP is exposed to multiple security threats. Conventional protection mechanisms do not fit into the highly dynamic, open and large-scale settings of VoIP infrastructures, and may significantly impact on the performance of such a critical service. We propose in this paper a runtime risk management strategy based on anomaly detection techniques for continuously adapting the VoIP service exposure. This solution relies on support vector machines (SVM) and exploits dynamic security safeguards to reduce risks in a progressive manner. We describe how SVM parameters can be integrated into a runtime risk model, and show how this framework can be deployed into an Asterisk VoIP server. We evaluate the benefits and limits of our solution through a prototype and an extensive set of experimental results.**

I. Introduction

Voice over IP (VoIP) has become a major paradigm for providing flexible telephony services while reducing infrastructure costs. The large-scale deployment of VoIP has been leveraged by the development of high-speed broadband access to the Internet. It has also been accelerated by the standardization of dedicated protocols, such as the SIP[1] signalling protocol. The term VoIP is often extended to cover other IP multimedia communications in general and convergent networks. VoIP services are much more open if compared to traditional telephony. A typical VoIP service is composed of three main parts: the user premises, the VoIP infrastructure for signaling and media transfer, and a number of supporting services. From a technical point of view, a SIP-based VoIP service is similar to an email service more than a conventional telecommunication service. Hence VoIP is expected to suffer from the same threats of the TCP/IP networks and services. VoIP therefore faces multiple security issues including vulnerabilities inherited from the IP layer and specific applicative threats. The attacks against VoIP include service disruption and annoyance, eavesdropping and traffic analysis, masquerading and impersonation, unauthorized access and fraud [1]. These attacks may have significant consequences on the telephony service, such as the impossibility for a client of making an urgent phone call.

The research community started to investigate the best ways of providing attack detection and protection for VoIP services. Researchers argue that intrusion detection is necessary to struggle against VoIP fraudsters. The main drawback of intrusion detection systems (IDS) is however their false positives and false negatives. Even a small rate of false alarms makes the use of an IDS unpractical if not impossible. When coming to the response, the prevention policy may have a significant impact on the availability and the performance of the service. For example, if annoying calls are coming from a peer VoIP provider, then blocking all the calls from that provider will ban the legal users from making calls to the under protection domain. A graduated and progressive treatment based on various counter-measures is required.

In that context we propose a risk management strategy coupling SVM anomaly detection with dynamic security safeguards. The objective is not in itself to define a detection method, but to show how this method can be integrated into a runtime risk model. This strategy aims at dynamically controlling the exposure of a VoIP infrastructure based on a set of graduated safeguards, in order to minimize the impact on the VoIP service performance. We argue that SVM have been already proven to be efficient and accurate in monitoring VoIP signaling traffic [2]. We integrate SVM parameters by extending a runtime risk modelling presented in [3]. We then describe how this solution can be deployed into a VoIP server and evaluate its performance though an extensive series of experiments.

The main contributions of this paper are: (a) the design of an architecture for supporting runtime risk management using support vector machines, (b) the specification of an anomaly detection model based on call detail records, (c) the extension of the rheostat runtime risk model [4] to take into account the SVM parameters, (d) the evaluation of our risk management strategy based on an implementation prototype and simulation results. In particular we will show how integrating risk management and anomaly detection permits to compensate the limits of both.

The rest of the paper is organized as follows. In Section II we expose our anomaly detection model based on support vector machines. Section III presents the runtime risk management strategy and the integration of SVM parameters. Section IV describes the deployment of this solution within a VoIP architecture, and the prototype implementation is detailed in Section V. In Section VI we detail experimental results and analyze the benefits in terms of cost and risk amplitudes. Related work is presented in Section VII. Finally Section VIII concludes the paper and identifies future work.

[1]Session Initiation Protocol

978-1-4244-8909-1/$26.00 ©2010 IEEE

This paper was peer reviewed at the direction of IEEE Communications Society by subject matter experts for publication in the CNSM 2010 proceedings.

II. VoIP anomaly detection

Anomaly detection consists in analyzing statistics (or features) provided by a monitoring system in order to reveal abnormal situations. The task of the monitoring system is to extract the pre-defined statistics from the raw data. The analysis of these statistics is based on a mathematical framework and is pre-empted by a training period where a model of normality is built. We describe in this section an anomaly detection model applied to VoIP networks and services.

A. Data sources in a VoIP infrastructure

Several data sources are available for performing anomaly detection in a VoIP architecture. These sources include:

- the network traffic, especially the protocols that are essential to the normal operation of VoIP calls (SIP, RTP, DNS). This is the typical data source for a network-based anomaly detection. We defined 38 statistics (or features) for the SIP traffic in [2].
- the log of VoIP servers and underlying operating systems. This is the typical data source for a host-based anomaly detection.
- the statistics provided by VoIP servers. In general these statistics depend on the internal design of these servers. For instance, the OpenSIPS[2] SIP proxy provides several groups of probes: core, memory, stateless statistics, transaction statistics, user location and registration.
- the call detail records (CDR). Monitoring this source of information is particularly important for fraud detection and SPIT treatment.

B. Mathematical framework

A rich set of machine learning algorithms may constitute a suitable framework for anomaly detection. We focus on Support Vector Machines (SVM) [5], which are known for their efficiency and accuracy in many application domains namely network-based and payload-based anomaly detection. SVM are also lightweight hence suitable for a runtime monitoring scheme. One-class SVM constitutes a geometric framework where the statistics are mapped into a feature space and anomalies are detected in sparsely populated regions. They are particularly suitable for unsupervised learning where clean data are difficult to obtain (as it is the case of VoIP).

The basic idea of one class SVM is to separate the points from the origin with the largest possible margin by means of a hyperplane. Alternatively, The hypersphere formulation suggests rather finding the smallest sphere enclosing the data points. The quarter-sphere formulation [6] is more adapted to typical IDS features which are one-sided on $\mathbb{R}_0^+$. The quarter-sphere problem resolution leads to find only the center of the sphere without the radius. It has proven that the center of the sphere converges to the mean of the data in this case. The radius plays the role of a threshold which can be used to control the specificity and the sensibility of the anomaly detection. The anomaly score of a point is the distance of this point to the center. The anomaly score is an important parameter in our approach since it determines the potentiality of an attack.

[2]http//www.opensips.org

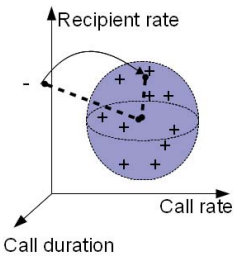

Fig. 1. SVM-based anomaly detection: when the source of the anomaly is omitted from the calculation, the corresponding point comes back to the normal region.

C. VoIP attack identification

Once an anomaly is detected, a response cannot be triggered if no information about the anomaly is provided. The attack identification may be based on one of these methods:

- If labeled data of the domain under one type of attacks are available, we can build a model of this type of attack, or a binary classification model (normal/attack type). The anomaly is identified to belong to this type of attack by comparing to one of these models. SVM also supports multi-classification.
- Each type of attacks has qualitative properties. For instance, SPIT calls are typically characterized by small call durations and a high rejection rate with respect to the normality model. We can test our anomaly data to have similar properties. Multiple attack types are identified through a tree of decision rules.
- Using specific visualization techniques such as the prediction sensitivity for quarter-sphere SVM [7]. The prediction sensitivity measures the degree to which prediction is affected by adding weight to a particular feature.

D. Attack source identification

The attack source identification is based on the fact that the suppression of their effects must move the anomaly point to lie again in the normal region. To give an example, let us assume that a list of CDRs is solely represented by the average call duration of all the calls. If the average call duration of a list of CDRs reveals abnormally small comparing to the normal average previously calculated during training, this situation is identified as abnormal. The cause of this abnormality is one or more sources that have generated the short-duration calls. If we suppress all the CDRs of one of these sources, the average of call durations must be closer to the training value and may be predicted as normal. This is described in the following algorithm:

1) range all the VoIP call sources by increasing order of call duration average,
2) suppress the top most ranked call source and add it to the list of suspicious sources,
3) recalculate the data point (new overall average of call durations),
4) test the data point with the anomaly detector,

5) if the new data point is predicted as normal then return the list of suspicious sources and exit, otherwise come back to the first step.

In general, we consider the anomaly score of a situation is composed of individual contributions of sources. The sources are then ranged by decreasing order of their scores. Eliminating the top ranked sources results in a normal prediction (Figure 1). The eliminated sources are considered as the cause of the abnormality. The mechanism of detection, classification and source identification can be visualized in 2D or 3D.

III. Runtime risk management

We propose to integrate the SVM parameters into an extended rheostat runtime risk management model capable of dynamically adapting the exposure of the VoIP service.

A. Rheostat runtime model

Given a system having a set of vulnerabilities $W = \{w_1, w_2, ...\}$, the risk is defined as the combination of two variables: (a) the probability that a threat $t_\alpha \in T$ transforms into a real attack by exploiting one of the vulnerabilities, and (b) the resulting impact on the system ($\mathcal{D}$) [8]. The probability of an attack occurrence is furthermore composed of two variables: the openness of the system $\mathcal{V}(t_\alpha)$ and the inherent potentiality (severity) of the attack $\mathcal{T}(t_\alpha)$. The potentiality of an attack is estimated by the anomaly detection system. The estimated risk is then defined by the following equation:

$$\mathcal{R} = \sum_{t_\alpha \in T} \mathcal{T}(t_\alpha) \times \mathcal{V}(t_\alpha) \times \mathcal{D}(t_\alpha) \quad (1)$$

Risk management consists in identifying threats, evaluating risks, and taking appropriate decisions to reduce them to an acceptable level [9]. Our schema relies on the extension of the rheostat risk model [4] which provides runtime support for dynamically controlling the exposure of a system. This control is driven by a cost-benefit analysis in order to provide an appropriate response. The exposure of the system is controlled by auxiliary security safeguards/checks applied in a progressive manner. The exposure $\mathcal{V}(t_\alpha)$ is defined by Equation 2 where $\hat{P}(t_\alpha)$ represents the set of operations that are required for performing an attack t_α.

$$\mathcal{V}(t_\alpha) = \sum_{o_\lambda \in \hat{P}(t_\alpha)} \frac{v(o_\lambda) \times s(o_\lambda)}{|\hat{P}(t_\alpha)|} \quad (2)$$

The initial exposure $v(o_\lambda)$ of a given operation o_λ is weighted by the $s(o_\lambda)$ factor quantifying the impact of the activation or the deactivation of security safeguards. This factor is set to 1.0 if no safeguard is activated and is set to 0.0 if the operation is fully controlled. The rheostat model addresses the trade-off between the openness of a system and its quality of service. In fact, each security safeguard is characterized by two attributes: (a) its ability to reduce the estimated risk and (b) its impact on the quality of service for the legal users.

The progressive risk management algorithm acts as follows: each time the estimated risk level bypasses a predefined threshold value, it selects the set of safeguards reducing the risk to an acceptable level and presenting the best cost-to-benefit ratio, as shown in Equation 3.

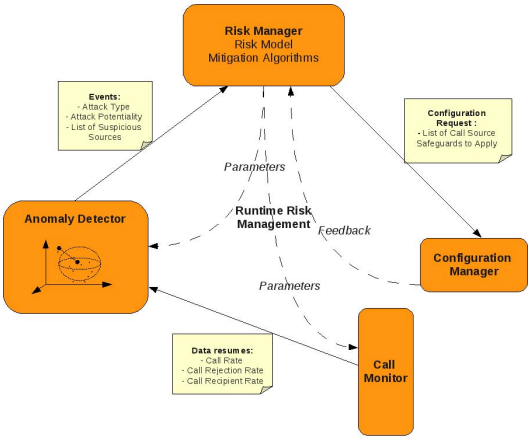

Fig. 2. Risk management architecture for VoIP infrastructures

$$minimize(\textstyle\sum_{o_\lambda \in \hat{P}(t_\alpha)} i(o_\lambda)) \quad and\ R_{new} \leq R_{threshold} \quad (3)$$

R_{new} corresponds to the risk level calculated after applying the security safeguards and $i(o_\lambda)$ quantifies their impact on the service performance. The relaxation algorithm permits to deactivate auxiliary safeguards in order to optimize the service performance when the risk level becomes unnoticed. The risk level of a threat t_α automatically decreases when no events corresponding to this threat has been observed for a given time period. At the expiration of an aging timer, the relaxation algorithm deactivates the safeguards. Further mathematical details about this modelling can be found in [3].

B. Extension based on anomaly detection

A management model such as the rheostat model is required to deal with the VoIP security threats, namely in areas like SPIT mitigation and fraud detection. Let us instantiate the model for the case of SPIT mitigation. SPIT calls may come from a peer VoIP provider hence blocking all the calls from the provider is not a practical solution. We define the risk level of a call source (containing both legal users and malicious users or botnets) as the number of SPIT calls that succeeds to reach the end-users hence annoying them. The risk level is composed of three variables: the intensity of the SPIT attack I (number of unsolicited calls per unit of time), the openness of the system (the set of security checks before forwarding the calls incoming from that source) and the annoyance at the end-user side which we assume constant for each successful SPIT call. We represent the openness for a call source (c) as the probability Pr_m that a malicious caller bypasses the set of safeguards (S) applied on this call source. We consider that the impact on the system is constant per successful SPIT call (cte). The SPIT risk level for multiple call sources ($c \in C$) is defined by Equation 4.

$$\mathcal{R}_{\text{SPIT}} = \sum_{c \in C} I_c \times Pr_m(S_c) \times \text{cte} \quad (4)$$

Applying a set of safeguards (S) for calls incoming from a call source imposes a cost for the legal calls incoming from the same source. This cost can be adding a delay time (d) to the call setup, or even a definite failure of it. We define Pr_h as the probability of a legal user to successfully bypass the set of safeguards. We define A as the cost for the failed legal calls. Let us consider multiple call sources $c \in C$, each generating N_c legal calls. The cost is then defined by Equation 5.

$$\mathcal{C} = \sum_{c \in C} N_c \times (Pr_h(S_c) \times d(S) + (1 - Pr_h(S_c)) \times A) \quad (5)$$

As previously mentioned in the model, our goal is to minimize the cost while reducing the risk to a low acceptable level. The reason of choosing different variables (Pr_h) and (Pr_m) characterizing openness for legal and illegal calls is that the legal calls are typically generated by humans and the unsolicited calls are typically generated by botnet machines. Examples of suitable safeguards for SPIT mitigation include: simulating a busy situation for the first call trial, requiring the resolution of a Captcha-like puzzle, making a Turing human/machine test, putting the caller in a waiting queue, and finally if nothing works, blocking the call. Our model can be extended to take into account the identity of the callee as well. For instance in an enterprise, reaching the director phone number should be harder than reaching one of his officers.

IV. Deployment architecture

We describe a deployment architecture based on an IPBX server, focusing on SPIT mitigation. The SPIT problem can be divided into two sub-categories: the inbound SPIT and the outbound SPIT. The outbound SPIT is generated by users registered to our VoIP service while the outbound SPIT is unsolicited incoming calls targeting our domain. The monitoring of the outbound SPIT is based on monitoring the calling profile of our users. In this context, each user account is considered as a call source. This approach cannot be scaled for inbound SPIT because we cannot build profiles for all possible external callers. Moreover, the attackers are easily able to change their identities. Instead, we suggest monitoring the IP addresses (or domain names) as call sources. The monitoring of the inbound SPIT must be based on the incoming calls of all the call sources. When the anomaly detector reveals an anomalous situation, the suspicious call sources are transmitted to the risk manager along with their respective potentialities. The risk manager decides to apply safeguards to each call source in function of its potentiality. The configuration system then put these decisions into action.

A. CDR-based statistics

In any case the monitoring must be based on statistics or features defined on the call history during a given period of time. The typical data source for our approach is the CDRs. We define a set of eight statistical parameters over a given list of CDRs such as:

- Call rejection rate: the SPIT calls may congest the victim VoIP network resulting in a large number of calls with end status as "Failed" or "Busy". A high call rejection rate indicates SPIT or flooding attacks.
- Average billing duration: The SPIT calls have generally shorter durations than normal calls. This is also because the end-users hang-up directly after recognizing a publicity message. We calculate the average duration over all the successful VoIP calls.
- Average call duration: the call duration is different than the billing duration since it contains the call setup time as well. For example in the SIP protocol, the billing duration is the time between the 200 OK and the BYE message while the call duration is between the INVITE and the BYE message. The average call duration of the failed calls may also be a good index of some anomalies.
- Call rate: the SPIT calls increment the overall call rate if compared to the normal call traffic.
- Context rate: the context of a call reveals the class of the dialed extension (e.g. local, department, international). This feature helps specially the detection of fraud and outbound SPIT.

These features are evaluated periodically at runtime. The time period between two successive evaluations is referred to as the monitoring window. The features are calculated over all the CDRs that have been added during the last period (i.e. all the calls that have ended). In training mode, we map each period to a point in the feature space. The set of the training points are used to generate the normality model. The training can also be done offline based on the previous CDR logs. In testing mode, each period is mapped to the feature space and predicted based on the training model. We suggest using a relatively small monitoring window (in the range of one minute) in order to detect the attacks at an early stage.

B. Architectural components

Our architecture aims to deploy the risk management model and the anomaly detection model into a VoIP PBX server. It is composed of four main functional components: the call monitor, the anomaly detector, the risk manager and the configuration manager. The call monitor is directly connected to the data source (the CDR database). It calculates the set of defined features based on a periodic schedule. It forwards the feature values in an appropriate format to the anomaly detector. The anomaly detector is responsible of revealing the attack situations and measuring their potentiality. The performances of the anomaly detector are necessarily limited in terms of sensitivity and specificity. The detection threshold is an important parameter to configure a good compromise between the sensitivity and the specificity of the anomaly detector.

The risk manager permits to deal with the intrinsic limits of the detector. The results of the anomaly detection are directly transmitted to the risk manager at an early stage (i.e. partial

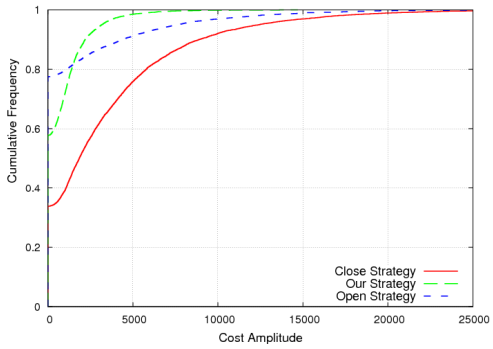

Fig. 3. Cost amplitude in comparison to other strategies

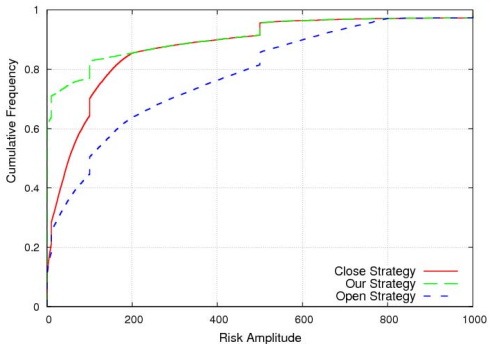

Fig. 4. Risk amplitude in comparison to other strategies

and low potentiality attacks). They are integrated progressively into the risk model in order to estimate the risk level at runtime. The risk manager integrates the progressive updating and relaxation algorithms for controlling the PBX dialplan in function of the estimated risk level. It is responsible of selecting a list of reactions when a situation is determined as highly risky. The configuration manager receives configuration requests from the risk manager. The configuration request contains a list of couples. Each couple represents a call source and a list of safeguards to be activated (or deactivated) for all the calls coming from this call source. The configuration manager acts at the dialplan level to protect the dialed extensions. In extension, the configuration manager sends feedback to the risk manager: for example indicators showing if a safeguard has been successfully applied or not, or experimented changes after setting a certain policy. Similarly, the risk manager is able to set certain parameters at the anomaly detector such as the detection threshold, and at the call monitor such as the monitoring window. The deployment architecture is depicted in Figure 2.

V. Prototype Implementation

We have implemented a prototype in an Asterisk[3]-based VoIP environment.We have used built-in Asterisk drivers and modules to connect to the MySQL database and to store the Call Detail Records (CDRs). The database is accessed by the monitoring package in order to query a list of CDRs based on a time interval and possibly other criteria. The CDR monitoring package has several functioning modes depending on three important settings:

- online/offline: In online mode, a monitoring window is defined (e.g. 5 minutes). At the end of each monitoring window, we query all the calls that have ended in this period. In offline mode, we query all the calls that have occurred between the start-time and the end-time of the required period.
- training/testing: In training mode, we extract the statistics and push them to the machine learning in order to build a normal profile. In testing mode, we extract the statistics and predict if they reveal a normal or an abnormal behaviors;
- individual/group/global: Depending on whether if we are monitoring a single source (e.g. IP, user account), a group of sources, or all the call activities we adopt our query request.

The anomaly detection algorithm is based on one-class SVM using the LibSVM library [10]. The value of the SVM decision function is taken as the anomaly score. The detection algorithm identifies the presence of SPIT or other abnormalities and reveals the list of potential actors. The identity of the actor is represented by the user account for a registered user and by the IP address for external calls. The monitoring package forwards the results to the risk management module.

The risk management module stores and manages the list of suspicious actors through the Asterisk database (AstDB). The Asterisk database is a simple implementation based on version 1 of the Berkeley database. We choose to use AstDB because it is simple and efficient to work with on real-time. The risk management module assigns a safeguard for each actor based on the risk management approach.

The extensions in the dial plan are protected by an AGI (Asterisk Gateway Interface) script. We coded our AGI script in Python using the AGI python toolkit. When an extension is called, the AGI script is run first. The AGI script takes the Asterisk channel parameters as arguments and looks in the Astdb to see if any safeguard has to be applied before calling the extension. We identify the caller by the *agi_channel* parameter. This parameter contains the caller's account name if the caller is a (registered) user of Asterisk or the IP address if the call is incoming from the outside. The AGI script exchanges data with Asterisk through Stdin, Stdout and Stderr pipes. Currently, our AGI script supports the following safeguards:

1) Responding with a busy message for the first call tentative,
2) Putting the call in hold for a random number of seconds,
3) Asking to dial a specific DTMF[4] tone in order to establish the call,

[3]http://www.asterisk.org/

[4]Dual-Tone Multi-Frequency

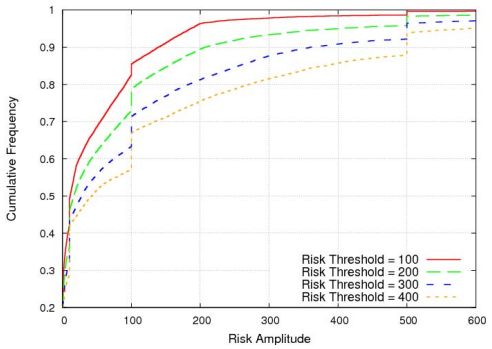

Fig. 5. Risk amplitude with different risk thresholds

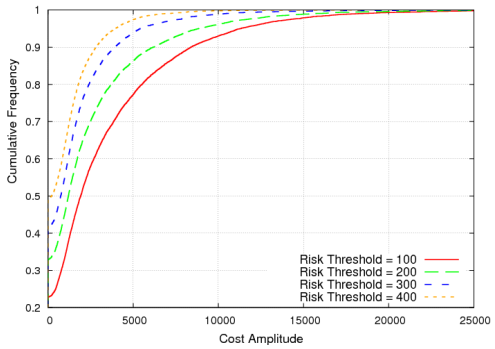

Fig. 6. Cost amplitude with different risk thresholds

4) Making the Answering Machine Detection (AMD) test in order to perform human/machine detection,
5) Redirecting the call to another destination (for instance, human filtering done by the secretary),
6) Blocking the call.

We have tested these safeguards against currently available SPIT attack tools such as Spitter/Asterisk[5], Warvox[6] and Voipbot[7]. The tests highlight the difference and complementary of safeguards in terms of both benefits and costs.

The lack of common labeled data in this domain does not help the comparison with other approaches. We want to provide a three-fold (traffic, call records, server statistics and logs) labeled data-set in order to fill this gap. We have already build a first package available online on the INRIA forge.

VI. Performance evaluation

We have conducted an extensive series of simulations in order to analyze the performances of our approach. The call arrival is represented by a Poisson law and a mean of 100 calls per unit of time. The call duration is represented by an exponential law and a mean of 10 seconds. The attacks are represented by 4 different types with increasing SPIT intensity (10, 100, 500 and 1000 SPIT calls per unit of time). We choose a constant value of 60 seconds for accounting the cost of failed normal calls. We define 5 different safeguards where each safeguard is characterized by three variables: delay, Pr_m (probability that a malicious call bypasses the safeguard) and Pr_h (probability that a "honest" call bypasses the safeguard. Pr_h is fixed for all safeguards as a uniform distribution in the $[0.8; 1]$ interval. The safeguards are ordered such as the first safeguard has the minimum induced delay (exponential distribution with $\lambda = 1/10$) and the maximum Pr_m (uniform distribution in the $[0.8; 1]$ interval). The last safeguard has the maximum induced delay (exponential distribution with $\lambda = 1/50$) and the minimum Pr_m (uniform distribution in the $[0; 0.2]$ interval). We define 5 different detection threshold with different (sensitivity, specificity) schemas. A smaller detection threshold means a higher sensitivity and a lower specificity. In result, a smaller detection threshold means a higher potentiality for a given attack intensity. These settings give us 30 different simulation scenarios (normal and different attack intensities vs. different detection thresholds). We make 10,000 Monte Carlo simulations per scenario, which provides an error term of less than 5% in our case. We use the same seed number for the pseudo-random number generation of all scenarios. Next, we expose a subset of our experimental results.

A. Comparison to other strategies

We compare our approach to two other strategies. The "Closed strategy" consists on using a strong safeguard (high Pr_m) as soon as a low potentiality is perceived. The "Open strategy" consists on applying a strong safeguard only when a high potentiality is perceived. Our strategy of risk management is to measure the potentiality, to estimate the risk level, to compare the estimated risk to our risk threshold, and in case to choose a suitable safeguard decreasing the risk to an acceptable level. The simulations show that our approach performs better in total.

As shown in Figures 4 and 3, the X-axis represents the amplitude of risk or cost. The Y-axis represents the cumulative frequency of simulations that lead to such an amplitude. More a curve is to the top-left side of the graph, more it represents good performance. The simulations show that our approach is better in terms of both cost and risk. However, the open strategy has a greater number of scenarios with zero or very small cost. The closed strategy has similar risks for scenarios of high potentiality. Notice the steps in the risk graph at 10, 100, 500 and 1000 amplitudes. These steps belong to the simulation runs where these attacks are perceived as acceptable (thus the risk level is not changed).

B. Impact of the risk threshold setup

The risk threshold parameter represents the system openness in our risk management model. A high risk threshold means that we prefer to minimize the cost in spite of a high risk level. A small risk threshold means that we accept to pay some cost in order to protect our system. Figures 5 and 6 depict four scenarios with different risk thresholds (100, 200, 300 and 400). Obviously, a risk threshold set to 100 provides the best performance in terms of risk and the worst in terms of cost. A

[5]http://www.hackingvoip.com/sec_tools.html
[6]http://warvox.org/
[7]http://voipbot.gforge.inria.fr/

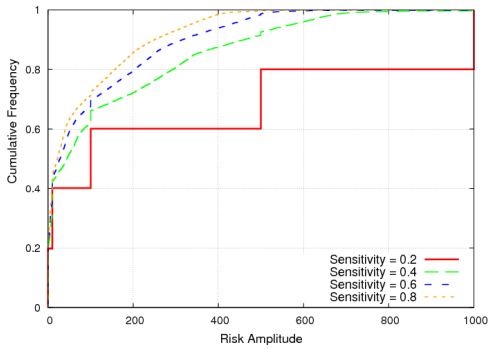

Fig. 7. Risk amplitude in function of detection sensitivity

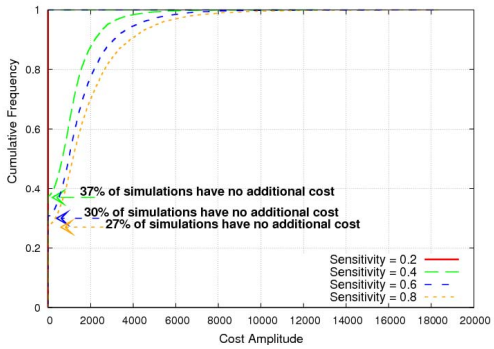

Fig. 8. Cost amplitude in function of detection sensitivity

risk threshold of 200 or 300 is a good compromise between the risk level and the imposed cost. The steps in the risk graph at 10, 100 and 500 amplitudes corresponds to the simulation runs where the respective attacks are perceived as acceptable. Thus no safeguard has been activated in order to mitigate the risk level.

C. Performance in function of detection sensitivity

It is important to study the performances of our approach in function of the anomaly detection characteristics. An intrusion detection system is characterized by two parameters: the sensitivity and the specificity which are generally dependent. By definition, the sensitivity is the rate of true positives or the abnormal time units that are properly detected. The specificity is the rate of true negatives or the normal time units that are properly detected.

In our case, the anomaly detector attributes a potentiality value to a situation rather than binary classifies it. In this context, the sensitivity is the probability of the anomaly detector to attribute a high potentiality value to a given attack situation. Respectively, the specificity is the probability to attribute low potentiality values when there is no attack. Figures 7 and 8 depict four scenarios with different sensitivity values (0.2, 0.4, 0.6 and 0.8) and a constant specificity value (0.8). At a specificity of 0.2 all the attacks are detected as tolerable resulting in high risk amplitudes. In contrast, no cost is paid for this scenario since no safeguards are applied. At the other extreme, with a specificity of 0.8, the risk level is minimized and we do not pay any cost for 37% of simulation runs.

D. Performance in function of detection specificity

The specificity parameter has no influence on the risk of the system. However, a bad specificity value leads to estimate a normal situation as risky, so to pay an unnecessary cost. Thus, we simulate only the case of no attack. Figure 9 depicts four scenarios with different specificity values (0.2, 0.4, 0.6 and 0.8). At a specificity equals to 0.8, we do not pay any cost since no normal situations are detected as attacks. An average specificity which is generally unacceptable for a real intrusion detection system (for instance a specificity equals to 0.6) preserves 24% of simulations from paying any cost. Also as shown, an anomaly detector having a very moderate specificity alters the performances of our approach.

VII. Related work

A few approaches really address risk management in VoIP networks and services, and most of them do not integrate an explicit risk model [11]. Existing work dealing with risk assessment in VoIP infrastructures includes methods for assessing threats (defender viewpoint) such as honeypot architectures and intrusion detection systems based on signatures, or based on anomalies [12], [13]. They also include methods for assessing vulnerabilities (attacker side) such as fuzzing-based discovery and auditing/benchmarking tools. For instance, in [14] the authors study the risk of call interceptions previously to the network deployment. They propose an attack tree and dependency graphs for identifying vulnerabilities. Besides, several work focus on the detection of VoIP attacks but without integrating a risk management approach. Risk models supporting the risk assessment phasis may be qualitative (based on linguistic scales), quantitative (based on probabilities) or mixed (based on aggregations of qualitative parameters) [8]. Existing work related to risk treatments permit to eliminate risks (risk avoidance) by applying best practices, to reduce and mitigate them (risk optimization) by deploying protection and prevention systems [15], to ensure against them (risk transfer) by subscribing an insurance contract or to accept them (risk retention) [16]. Work on IT change management such as [17], [18] are considered as out of our security-oriented scope.

The SPIT problem is extensively studied because of its importance for the future of VoIP. The key issue with SPIT identification is the caller identity. Quittek et. al. [19] apply hidden Turing tests on the caller side and compare their results to typical human communication patterns. For passing these tests, significant resource consumptions at the SPIT generating side would be required which contradicts the spammer's objective of placing as many SPIT calls as possible. A survey of protection techniques against SPIT is given in [20]. The authors argue in favour of combining complementary techniques, which is in coherence with our approach. VoIP SEAL [15] implements a two-stage decision process: The first stage contains modules which analyze a call only by

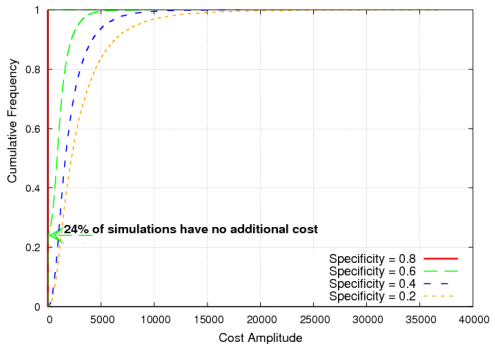

Fig. 9. Performance in function of detection specificity

looking at information which is available before actually answering the call. The second stage consists of modules which actually interacts with the caller or the callee to refine the detection. Since the second stage modules introduce some inconvenience, a scoring system is deployed at the first stage to determine if they will be used or not. Rather than Turing tests, other modules include white/black list, simultaneous calls, call rate, and URI's IP/domain correlation. Finally, the end-user feedback is taken into account if the SIP-client is instrumented for that. This work is the most similar to our approach but does not explicitly propose a risk management model.

VIII. Conclusions and Future Work

Telephony over IP is a critical service exposed to multiple security threats. Protection mechanisms exist but may seriously deteriorate the service performance. Applying risk management methods and techniques to VoIP infrastructures provide new opportunities for addressing the trade-off between security and quality of such sensitive services. In this context, we have considered a runtime risk management solution using SVM-based anomaly detection for automatically and continuously adapting the exposure of the VoIP service. The exposure is controlled by the activation and deactivation of safeguards in a graduated manner. The detection schema is based on machine learning namely support vector machines for identifying and measuring the abnormal deviations of learnt call patterns. We have proposed a deployment architecture based on an IPBX server and composed of four functional components: call monitor, anomaly detector, risk manager and configuration platform. Our architecture extends the rheostat formal risk model and identifies a subset of safeguards for SPIT mitigation. Finally, we have evaluated the performances of our solution through an implementation prototype and an extensive set of simulations. In results, the integration of the risk model with SVM-based anomaly detection, clearly contributes to a more appropriate response to the possible threats. Additional experiments in a real enterprise network are envisioned to complete these results from a practical viewpoint, but such a deployment is not easily authorized because of user privacy statements.

As future work we are planning to extend our approach in order to cover a larger scope of VoIP threats. The objective is to specify a unified strategy for dealing with multiple security attacks in these environments. We would also like to refine our risk model and supporting a dynamic feedback scheme between the components of our architecture. In particular, we are interested in developing self-configuration mechanisms for setting the risk model parameters in an adaptive manner. Finally, we will investigate new opportunities offered by financial engineering techniques for improving the selection of security safeguards.

References

[1] Voice over IP Security Alliance, "VoIP Security and Privacy Threat Taxonomy," *www.voipsa.org/Activities/taxonomy.php*, October 2005.

[2] M. Nassar, R. State, and O. Festor, "Monitoring SIP Traffic Using Support Vector Machines," in *Proceedings of the 11th International Symposium on Recent Advances in Intrusion Detection (RAID '08)*. London, UK: Springer-Verlag, 2008, pp. 311–330.

[3] O. Dabbebi, R. Badonnel, and O. Festor, "Automated Runtime Risk Management for Voice over IP Networks and Services," in *Proc. of the 12th IEEE/IFIP Network Operations and Management Symposium (NOMS 2010)*, Osaka, Japan, April, 2010.

[4] A. Gehani and G. Kedem, "RheoStat: Real Time Risk Management," in *Proc. of the 7th International Symposium on Recent Advances in Intrusion Detection (RAID 2004)*, 2004.

[5] V. Vapnik, *Statistical Learning Theory*. Wiley and Sons, 1998.

[6] P. Laskov, C. Schäfer, and I. Kotenko, "Intrusion detection in unlabeled data with quarter-sphere support vector machines," in *Proceedings of the first Conference on Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA)*, 2004, pp. 71–82.

[7] P. Laskov, K. Rieck, C. Schäfer, and K.-R. Müller, "Visualization of anomaly detection using prediction sensitivity," in *Sicherheit*, 2005.

[8] T. Bedford and R. Cooke, *Probabilistic Risk Analysis: Foundations and Methods*. Cambridge University Press, April 2001.

[9] M. Modarres, *Risk Analysis in Engineering: Techniques, Trends, and Tools*, Mohammad Modarres, Ed. Taylor & Francis Editions, 2006.

[10] C. Chang and C. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[11] O. Dabbebi, R. Badonnel, and O. Festor, "Managing Risks at Runtime in VoIP Networks and Services," in *Proc. of the Autonomous Infrastructure, Management and Security (AIMS'10, PhD Workshop)*, June 2010.

[12] R. Dantu, P. Kolan, and J. W. Cangussu, "Network Risk Management using Attacker Profiling." *Security and Communication Networks*, vol. 2, no. 1, pp. 83–96, 2009.

[13] D. Shin and C. Shim, "Progressive Multi Gray-Leveling: A Voice Spam Protection Algorithm," *IEEE Network Magazine*, vol. 20, Sep. 2006.

[14] M. Bunini and S. Sicari, "Assessing the Risk of Intercepting VoIP Calls," *Elsevier Journal on Computer Networks*, May 2008.

[15] R. Schlegel, S. Niccolini, S. Tartarelli, and M. Brunner, "Spam over Internet Telephony (SPIT) Prevention Framework," in *Proc. of the IEEE Global Communications Conference (GLOBECOM'06), San Francisco, USA*, November 2006.

[16] ISO/IEC 27005, "Information Security Risk Management, International Organization for Standardization, http://www.iso.org," June 2008.

[17] A. Keller, J. Hellerstein, J. Wolf, K. Wu, and V. Krishnan, "The CHAMPS System: Change Management with Planning and Scheduling," in *Proc. of IEEE/IFIP Network Operations and Management Symposium (NOMS'04)*, April 2004.

[18] J. Wickboldt, L. Bianchin, R. Lunardi, F. Andreis, R. L. dos Santos, B. Dalmazo, W. Cordeiro, A. R. de Sousa, L. Z. Granville, L. P. Gaspary, and C. Bartolini, "Computer-Generated Comprehensive Risk Assessment for IT Project Management," in *Proc. of 20th IFIP/IEEE International DSOM Workshop (DSOM'09)*, October 2009.

[19] J. Quittek, S. Niccolini, S. Tartarelli, M. Stiemerling, M. Brunner, and T. Ewald, "Detecting SPIT calls by checking human communication patterns," in *IEEE International Conference on Communications (ICC 2007)*, June 2007.

[20] V. M. Quinten, R. van de Meent, and A. Pras, "Analysis of Techniques for Protection Against Spam over Internet Telephony ," in *Proc. of 13th Open European Summer School EUNICE 2007*, July 2007.