Securing IMS against Novel Threats

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

[1] Bell-Labs Germany, Alcatel-Lucent Deutschland AG, Lorenzstrasse 10, 70435 Stuttgart, Germany
[2] Fraunhofer Institute FIRST, Intelligent Data Analysis, Kekulestrasse 7, 12489 Berlin, Germany
[3] Technical University of Berlin, Machine Learning Group, Franklinstr. 28/29, 10587 Berlin, Germany

This is a preprint of an article published in the Bell Labs Technical Journal, 14(1), p.243–257, 2009.
http://www.interscience.wiley.com

Abstract

Fixed mobile convergence (FMC) based on the 3GPP IP Multimedia Subsystem (IMS) is considered one of the most important communication technologies of this decade. Yet this all-IP-based network technology brings about the growing danger of security vulnerabilities in communication and data services. Protecting IMS infrastructure servers against malicious exploits poses a major challenge due to the huge number of systems that may be affected. We approach this problem by proposing an architecture for an autonomous and self-sufficient monitoring and protection system for devices and infrastructure inspired by network intrusion detection techniques. The crucial feature of our system is a signature-less detection of abnormal events and zero-day attacks. These attacks may be hidden in a single message or spread across a sequence of messages. Anomalies identified at any of the network domain's ingresses can be further analyzed for discriminative patterns that can be immediately distributed to all edge nodes in the network domain.

1 Introduction

The IP Multimedia Subsystem (IMS) "is a global, access-independent and standards-based Internet Protocol (IP) connectivity and service control architecture that enables various types of multimedia services to end users using common Internet-based protocols" [15]. It was originally standardized by the 3rd Generation Partnership Project (3GPP[1]) and later adopted and extended by the European Telecommunications Standards Institute (ETSI) Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN). A converged fixed mobile convergence (FMC) network [23] distinguishes three planes, as shown in Figure 1. The IMS plane performs the control/signaling layer functions; it has open interfaces to the IP transport plane, and to the service and application plane. IMS technology relies on principles and protocols of the Internet Engineering Task Force (IETF). The Session Initiation Protocol (SIP) plays the key role in controlling diverse multimedia services. The SIP protocol enables one to embed conversational services into numerous new applications for end users across multiple mobile and fixed access networks.

From the security point of view, the FMC architecture must be protected against threats from multiple communication domains and protocols including SIP, Media Streaming Real Time Transport Protocol (RTP), and Internet Protocol (IP). The focus of this paper lies on SIP layer security. All of the three classic security goals—authenticity, availability, and integrity—can be attacked by exploiting vulnerabilities in the SIP stack implementations running on IMS edge or core nodes. By compromising an IMS node, an attacker may carry out an array of further attacks on IMS services, ranging from the loss of confidentiality to gross monetary abuse, e.g., by assuming a false identity and tampering with the call accounting information. Major damage to availability can be inflicted by crashing SIP stacks using irregular protocol requests.

The above mentioned risks inherent to a wide-scale deployment of IMS technology require a careful and multi-faceted consideration of security issues. These

[1] 3GPP is a trademark of the European Telecommunications Standards Institute.

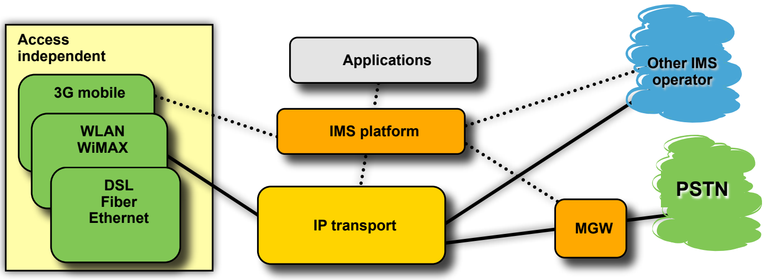


Figure 1: IP-based FMC architecture. Abbreviations 3G–Third generation, DSL–Digital subscriver line, FMC–Fixed mobile convergence, IMS–IP Multimedia Subsystem, IP–Internet Protocol, MGW–Media gateway, PSTN–Public switched telephone network, WiMAX–Worldwide Interoperability for Microwave Access, WLAN–Wire local area network.

issues necessarily encompass preventive as well as reactive instruments. Preventive tools, such as encryption, authentication, and integrity control, represent fundamental building blocks without which secure operation of IMS is unthinkable. While they are important design factors, these methods alone, however, cannot fully guarantee the security of IMS infrastructures. Implementation errors, configuration problems, and subtle semantic vulnerabilities are inevitable in such infrastructures due to their complexity. To prevent malicious exploitation of IMS, preventive techniques should be complemented by reactive mechanisms whose goal is to provide an adequate response to security problems that cannot be solved by design alone.

Most of the previous work on SIP security has focused either on preventive measures or on the response to known threats. It is, however, clear that in view of the exploding variability of currently observed malicious software, IMS infrastructures must be equipped with a capability to detect previously unknown attack variants. Such capability can only be provided by self-learning components which capture characteristics of observed normal data to flag anomalies as malicious events. Such methods have been previously proposed for general-purpose intrusion detection systems for IP networks, e.g., [8,10,16,17,26]. The peculiarities of the network protocols used in IMS, for example the Session Initiation Protocol, necessitate the development of specialized intrusion detection techniques tailored to their semantics.

The main contribution of this paper is a new security architecture for IMS containing mechanisms for detection and response to unknown attacks. The heart of this architecture is an anomaly detection component, which is deployed alongside the typical components of a SIP stack. The anomaly detection component assesses the irregularity of incoming SIP messages, the rationale being that unusual events require more careful treatment in the remaining processing chain than normal ones. Anomaly scores generated in our architecture can be used for intelligent scheduling of SIP messages as well as for generation of signatures for anomalous events. Such signatures can be further deployed on a domain scale in order to share information between various edge nodes. Experimental evaluation of SIP traffic obtained from several real next-generation network (NGN) laboratories, and synthetic attacks generated by a popular fuzzer developed by Codenomicon, has demonstrated that the proposed architecture can reliably detect (with up to 99 percent accuracy) previously unknown attacks at IMS infrastructures.

2 Attack Categories and State-of-the-Art Solutions

A large variety of potential attacks are threatening IMS core networks. This section concentrates mainly on threats that are addressing the IMS control layer, specifically different types of call session control function (CSCF), as well as the application servers (AS) above the control layer. When categorizing the different types of attacks and threats, it is useful to distinguish between time dependent and time independent attacks [22].

Time dependent attacks aim to exhaust resources at different layers, from link bandwidth up to the computational resources at IMS control and AS. Flooding attacks are specialized time dependent attacks and can occur on different layers. TCP/SYN flooding, Smurf attacks, and SIP message flooding are examples of flooding attacks. The SIP message flooding attacks detailed below aim to overwhelm IMS control layer functions in the following ways:

- *REGISTER flooding attacks* generate a huge number of REGISTER messages with different spoofed and faked source addresses which can overwhelm the IMS control layer functions.
- *INVITE flooding attacks* are similar to REGISTER flooding, but these attacks try to break the IMS authentication process.
- *INVITE and REGISTER* response flooding attacks attempt to obtain valid authentication or authorization credentials.

Time independent attacks belong to the second category of attacks. These attacks insert their threats with each message, thus affecting their victim instantly. Structured query language (SQL) injection and time independent SIP message flow attacks are assigned to this category.

- *SQL attacks* try to perform data modifications and may even disturb database services.
- *Time independent SIP message flow attacks* try to either tear down sessions (BYE attack), terminate pending sessions (CANCEL attacks), modify sessions (Re-INVITE attacks), or eavesdrop on sessions (REFER attacks).

In the future, time independent attacks may be envisioned which are only visible if multiple messages are taken into account. Besides the IMS threats listed above, the following attacks must also be taken into account:

- *Eavesdropping and password guessing.* This approach tries to obtain session information to generate hijacking-style attacks
- *Registration and session hijacking.* Here, an attacker tries to register on another user's behalf and forward a session via the attacker's device.
- *Man-in-the-middle.* These attacks involve interception and modification of message flows.

3GPP standard has specified security solutions for IMS which consist of authentication and key agreement as well as integrity and confidentiality mechanisms to secure SIP messages. Still, these functionalities do not completely secure the IMS core and/or application servers against SIP message flooding, SIP message flow, fuzzing, and SQL injection. To further improve the protection against application layer attacks, intrusion detection/prevention solutions have been introduced. These intrusion detection and prevention solutions perform anomaly detection or misuse detection. In the case of anomaly detection, they are applying algorithms that investigate the normal behavior of computer systems. This method has the advantage of detecting unknown attacks but suffers from high false-positive results. In the case of the misuse detection mechanism, the detection algorithm is supplied with all known attack descriptions. This method therefore has a very low false positive rate, but it can only detect previously described attacks.

3 IMS Network Environment and Basic Architecture

A typical deployment scenario of an NGN foresees the placement of session border controllers (SBCs) at the border of the core network towards the access networks as well as to peering networks, as shown in Figure 2. With the advancement towards IMS/TISPAN networks, SBCs are available on the market which offer selected IMS functionality, for example, a proxy call session control function [14].

To fulfill enhanced security, availability, and integrity requirements, it is desirable that a major part of the security functionality is located at the ingress ports of the domain, and herewith at the ingress ports of the border nodes (SBC). The goal is to shield the core IMS functions from any signaling threats coming from the outside. The detailed security architectures discussed in the following sections are located at the ingress ports of these SBCs.

The basic SIP reactive security subsystem consists of multiple processing stages, each performing dedicated functions at different layers. A combination of stateless and stateful functionalities performs identification for different kinds of misuse. Figure 3 shows the four-step pipeline processing of this basic security architecture whereby the processing layer steps up from left—the layer 3 and layer 4 (L3/L4) firewall

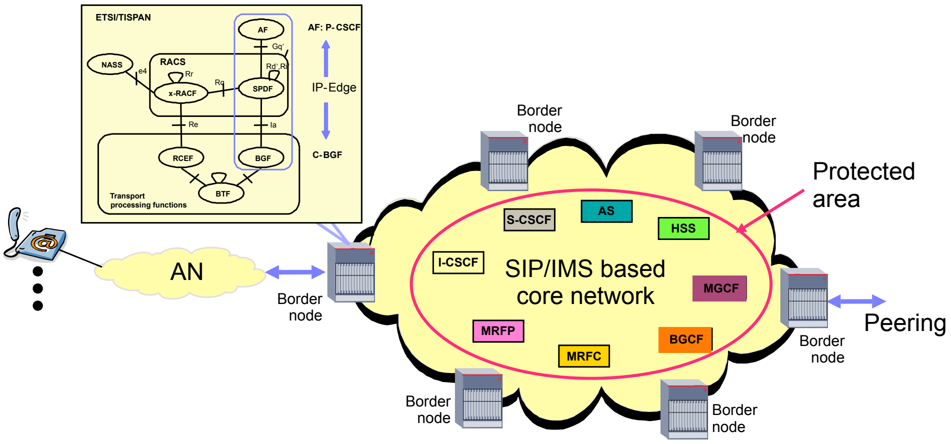


Figure 2: Core network with border nodes at the edges. Abbreviations: AF–Application function, AN–Access network, BGCF–Border gateway control function, BGF–Border gateway function, C-BGF–Core-BGF, CSCF–Call session control function, ETSI–European Telecommunications Standard Institute, HSS–Home subscriber server, I-CSCF–Interrogative CSCF, IMS–IP Multimedia Subsystem, IP–Internet Protocol, MGCF–Media gateway control function, MRFC–Media resource function controller, MRFP–Media resource function processor, NASS–Network attachement subsystem, P-CSCF–Proxy CSCF, RACF–Resource and admission control function, RACS–Resource and admission control subsystem, RCEF–Resource control enforcement function, S-CSCF–Serving CSCF, SIP–Session Initiation Protocol, SPDF–Service-based polidy decision function, TISPAN–Telecommunications and Interner converged Services and Protocols for Advanced Networking.

function—to right—the SIP stack at the application layer. The computational effort spent per SIP message increases in line with the degree of semantic message comprehension from left to right. Feedback links from higher layer to lower layer processing steps have been introduced to empower higher layer functional blocks, which are capable of identifying more subtle attacks, to control the behavior of the preceding functions. In the case of identified misuse, one of the higher layer processing steps can initiate an early and selective blocking or dropping on a per message, session, client, or IP address/port basis. As a result, the offered SIP message load can be controlled before reaching the SIP stacks and thus fulfilling stringent requirements on computational performance.

The first functional block of the architecture after the ingress port is a firewall. On one hand, its major task is to protect the subsequent functional modules of the border node from any kind of well-known layer 3 and layer 4 attacks, such as IP packet fragmentation, time dependent layer 3 and layer 4 attacks, e.g., Internet ICMP/TCP/UDP floods, smurf attacks, or denial of service (DoS) attacks. On the other hand, the SIP stack and/or the session queue unit can control the firewall to open or close dedicated firewall ports for a specific time period. Herewith massive SIP message attacks can be discriminated compared to well-behaving clients.

The firewall forwards a signaling message to a SIP pre-processing unit, which performs first SIP conformance validation, applying, for example, an augmented Backus-Naur form (ABNF) checker. After protocol conformance is verified, the sorting of the header fields is performed. The sorting order is determined by a typical usage frequency by the different CSCFs and application servers, so that none of the CSCF and AS have to scan the entire SIP message to gather all the relevant information [3,27]. Another important pre-processing function normalizes variations of allowed header field names into a consistent name (e.g., “FrOm” or “F” or “FRoM” or others into “FROM”). A complementary design of the SIP pre-processing, session queuing, and parsing subfunction of the SIP stack leads to a further increase in throughput. Hereby, the components make use of a shared memory with a clearly specified mes-

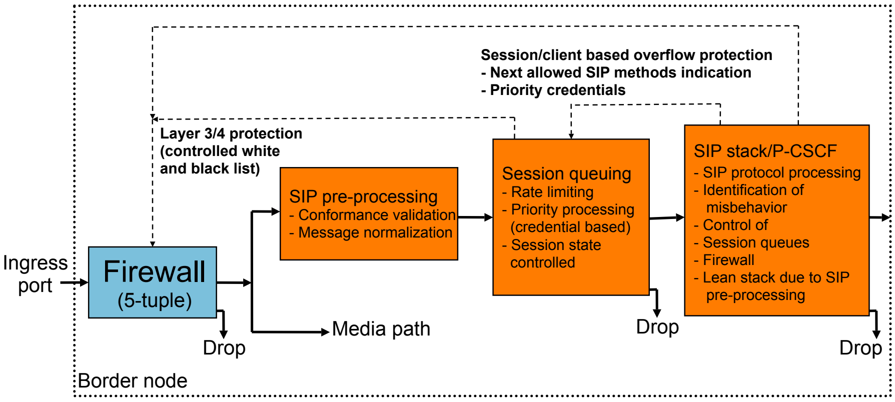


Figure 3: Basic SIP reactive security system. Abbreviations: CSCF–Call session control function, P-CSCF–Proxy CSCF, SIP–Session Initiation Protocol.

sage map of the messages. The SIP pre-processing unit stores the major headers of the SIP message at a specified location in a message memory map. As soon as the SIP pre-processing has extracted characteristic message headers, it indicates this to the session state management and message rate supervision within the session queuing unit. The session queuing unit provides early feedback on whether the session queue will drop the current message. In that case, the SIP pre-processing unit would receive a drop indication, and would immediately stop any further pre-processing activity and release the allocated memory.

The tasks of session queuing are manifold. The first priority is to manage all the sessions for this ingress port within logical queues. This includes instantiations, releases of session queues, and the supervision of the instantiation rate, but it also includes the message rate supervision per individual queue, whereby the message rate depends on several aspects. These aspects are determined by the session activity of the clients and the overall system load.

After the SIP stack has processed a SIP message, it provides feedback information to the session queuing module indicating whether the message was compliant (through compliance certificates) according to the session state; additionally, it indicates the set of subsequent message types allowed. The compliance certificates are used by the session queues to prioritize all correctly-behaving clients in high load situations. A message-selective gatekeeper function at the entrance of the session queues ensures that the only message types accepted are those allowed access according to the feedback information. The above described functionalities guarantee that only syntactically correct, rate supervised, and session state conforming SIP messages can allocate the valuable application layer processing of the SIP stack(s).

The session queuing module and multiple parallel running SIP stack processes create a single queue multiple server system. The next free SIP stack process will receive the next waiting message. This is achieved by providing the SIP stack at task start with memory pointers into the message record as well as into the session state memory. During SIP message processing, the SIP stack modifies the session state memory and can change the message record.

The basic architecture described provides a key functionality to ensure reliable operation of an IMS edge node under various load conditions. This functionality allows detection of many forms of known misuse, e.g., flooding or other well-known attacks. But with the increasing dissemination of VoIP and IMS systems, the number of security threats and the occurrence of new and unknown threats will greatly increase. Therefore, an additional functionality is required to handle new and unknown attacks. The following section presents an anomaly detection component for IMS edge nodes that is capable of detecting novel attacks with convincingly low false-positive rates. After presenting the basic anomaly detection component and the results of its experimental evaluation, additional features of the component, such as automatic signature generation, are described.

4 Anomaly Detection

Anomaly detection methods aim at identifying abnormal events based on a learned model of normality. The majority of anomaly detection methods are defined for vectorial data, and thus are not directly suitable for application to IMS protection. To address this issue, we derive a technique for embedding SIP messages in a vector space, which reflects typical characteristics of observed traffic patterns.

4.1 Feature Extraction

The anomaly detection approach builds on the vector space model, a generic method for feature embedding commonly used in the domain of information retrieval [6,20]. A document—in this case a SIP message—is characterized by occurrence frequencies of contained feature strings, S. Given a string $s \in S$ and a SIP message x, one determines the number of occurrences of s in x and thus obtains a frequency value $f(x,s)$. The frequency of s acts as a measure of its importance in x, e.g., $f(x,s) = 0)$ indicates no importance, while $f(x,s) > 0$ reflects the specific contribution of s to x. An embedding function m is derived by mapping SIP messages to an $|S|$-dimensional vector space spanned by the frequencies of all strings in S. The function m is defined as

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

where X denotes the set of all possible SIP messages and $\mathbb{R}^{|S|}$ a vector space over the real numbers. In contrast to textual documents, however, we cannot define a feature set S a priori, as patterns of unknown and novel attacks are impossible to determine in advance. To solve this problem the set of feature strings S is defined implicitly using the definitions of "tokens" and "n-grams".

In an implicit view, tokens correspond to all possible strings separated by specific delimiter symbols. If we denote all byte values by B and define D as delimiter symbols, a set S referred to as *tokens* is given by $S := (B \backslash D)^*$, where * is the Kleene closure corresponding to all possible concatenations. The granularity of this feature extraction can be controlled using the delimiter set D. The fewer delimiters are defined, the more specific are the extracted tokens. The following example illustrates how a SIP message is mapped to a vector space using the notion of tokens, where the set of delimiters is $D = \{\square, \texttt{@}, \texttt{:}, \texttt{/}\}$.

$$m(\texttt{BYE}\square\texttt{SIP:JOHN@DOE}\square\texttt{SIP/2.0}) \mapsto \left(\begin{array}{c} 1 \\ 2 \\ 1 \\ 1 \\ 1 \end{array}\right) \begin{array}{c} \texttt{BYE} \\ \texttt{SIP} \\ \texttt{JOHN} \\ \texttt{DOE} \\ \texttt{2.0} \end{array}$$

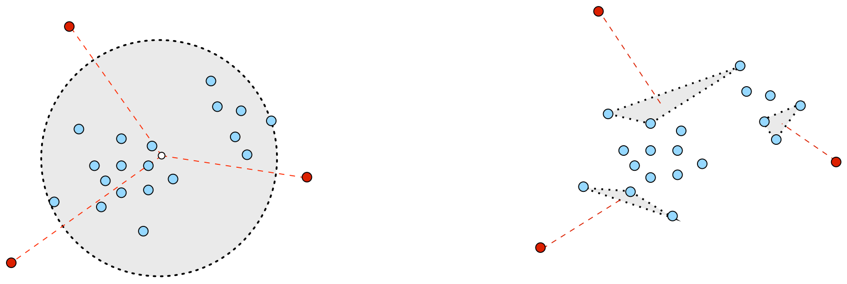
Tokens are intuitive and expressive to the human analyst, yet they may not always identify novel threats, due to the definition of delimiter symbols in advance. An alternative technique for implicit definition of feature strings S is the extraction of n-grams. Feature strings are extracted by moving a sliding window of length n over the SIP message. 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 a simplified SIP message yields the following feature vector

$$m(\texttt{BYE}\square\texttt{SIP:JOHN@DOE}\square\texttt{SIP/2.0}) \mapsto \left(\begin{array}{c} 1 \\ 1 \\ 2 \\ 2 \\ \vdots \end{array}\right) \begin{array}{l} \texttt{BYE}\square \\ \texttt{YE}\square\texttt{S} \\ \texttt{E}\square\texttt{SI} \\ \square\texttt{SIP} \\ \quad\vdots \end{array}$$

The vector space induced by tokens and n-grams is high-dimensional, e.g., for $n = 4$ there exist 256^4 different dimensions. Computing and comparing vectors in such high-dimensional spaces seems infeasible at a first glance. However, for both types of features, the number of feature strings contained in a single SIP message is linear in its length. This sparse representation of the embedding can be exploited to derive linear-time methods for extraction and comparison of feature vectors [18], which ultimately enables efficient anomaly detection over embedded SIP messages.

4.2 Anomaly Detection

The proposed embedding function m maps SIP messages into a vector space in which various learning algorithms can be applied for anomaly detection. We herein concentrate on two simple methods based on geometric models of normality: a *global* and a *local* anomaly detection. The basis for such geometric



(a) Global model

(b) Local model

Figure 4: Global and local anomaly detection.

learning models is a distance function d, which assesses the dissimilarity of two messages x and z by $d(x,z) = ||m(x) - m(z)||$ and corresponds to a Euclidean distance in the vector space. Messages originating from a similar context yield small distances and lie close to each other, while messages from different contexts result in greater distances.

Based on this notion of a distance for SIP messages $X = \{x_1, \ldots, x_n\}$, a global model for anomaly detection is defined by placing a hypersphere around a set of given SIP messages embedded into a vector space. In particular, one seeks the smallest representation of X corresponding to the hypersphere with minimum volume, which can be determined by solving the following optimization problem:

$$\mu^* = \underset{\mu}{\operatorname{argmin}} \max_{1 \le i \le n} ||m(x_i) - \mu||,$$

where μ^* is the center of the optimal hypersphere. Figure 4 depicts global anomaly detection. As an example, Figure 4(a) shows a set of points enclosed by a two-dimensional sphere with minimum volume. Anomalies lie outside the sphere due to their large distance from the center. 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 âĂIJsoftensâĂİ the margin of the hypersphere, such that outliers and unknown attacks can be compensated. An introduction to this regularized learning model is provided in [9] and [25], which cover the respective theory as well as the efficient computation.

Once the center μ^* of the smallest hypersphere has been found, the deviation g from this global model of normality is determined by computing the distance of an incoming SIP message z from μ^*. Formally, this distance is defined using the vectors μ^* and $m(z)$ in the feature space spanned by frequencies of n-grams or tokens as follows

$$g(z) = ||m(z) - \mu^*||.$$

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

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. To detect anomalies in such settings, we introduce a local anomaly detection scheme which assesses deviation of a message by considering only a fraction of messages in the training data. The model is derived using the notion of k-nearest neighbors. We define the neighbors of a vector $M(z)$ using a permutation p of X, such that the embedded message $m(x_{p[i]})$ is the i-th nearest neighbor to z in terms of distance. The local deviation l is then computed by

$$\begin{aligned} l(z) = & \frac{1}{k} \sum_{i=1}^{k} ||m(z) - m(x_{p[i]})|| \\ & - \frac{1}{k^2} \sum_{i=1}^{k} \sum_{j=1}^{k} ||m(x_{p[j]}) - m(x_{p[i]})||. \end{aligned}$$

Messages strongly deviating from their k-nearest neighbors yield a large average distance, while messages close to their neighbors get a low deviation score. In particular, the first term emphasizes points that lie far away from its neighbors, whereas the second term discounts abnormality of points in wide neighborhood regions. Figure 4(b) illustrates local anomaly detection, where the anomalies are identified by a large distance to neighboring points. In contrast to the global model, local anomaly detection requires determining several distance values. For each incoming message $O(nk^2)$ distance computations need to be performed for finding the k-nearest neighbors and calculating l.

To demonstrate the capabilities of the proposed detection methods, we conducted experiments on realistic SIP traffic and attacks. For our experiments we generated an evaluation data set comprising 4,428 SIP messages and 10,000 attacks. These SIP traces contain contiguous and interleaved SIP dialogs recorded at a network edge ingress. The messages originated from several NGN test labs and research setups where multiple services and inter-working tests are performed. 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[2] 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 attacks are post-processed to eliminate any remaining redundancy by permuting the sequence of the header fields and randomizing certain header and parameter values.

Figure 5 depicts the detection performance of the anomaly detection methods averaged over independent samples of the evaluation data using 2-grams, 4-grams, and tokens as string features. Figure 5(a) shows results for the global anomaly detection method and Figure 5(b) for the local anomaly detection method. The performance is presented as receiver operating characteristic (ROC) curves, which show the false positive rate of a method 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.

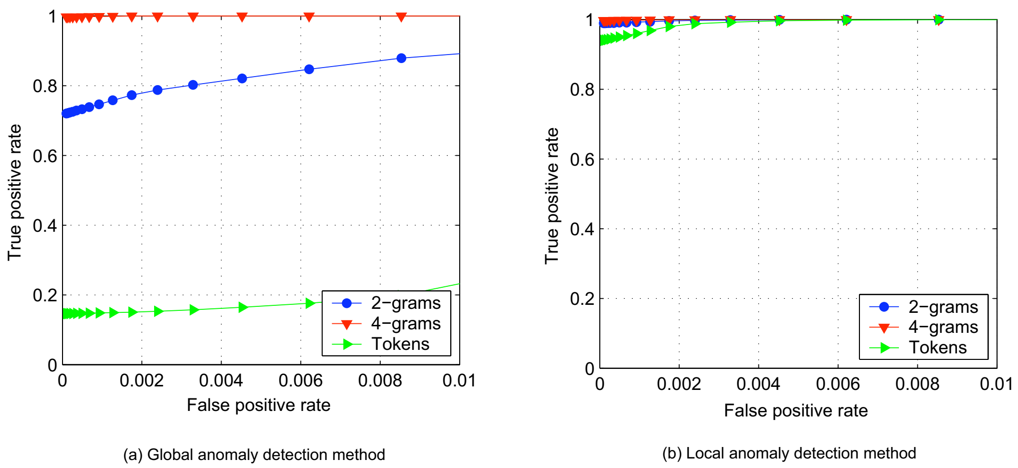
The local anomaly detection method yields significantly higher detection accuracy in comparison to the global detection method. In particular, for all types of feature strings a true-positive rate over 97 percent is achieved with no false positives. Moreover, for the 4-grams features, over 99 percent of the attacks are detected—even though all attacks were unknown to the system during application. For the global anomaly detection method only the 4-gram features enable similar accuracy and in contrast, the token features provide a very poor performance. The embedding to a vector space using 4-grams hence enables a very effective discrimination of normal traffic and attacks by capturing particular substrings related to normal or anomalous messages.

4.3 Signature Generation

An additional step towards integration of anomaly detection methods into practice is inter-linkage with existing signature-based detection systems. Once a novel attack has been identified by anomaly detection, a further step is to automatically generate a corresponding attack signature and distribute it to signature-based detection systems in the IMS domain. However, deriving a signature from a single anomaly is cumbersome and thus, as a preprocessing step, detected anomalous messages are grouped into attack clusters using algorithms such as linkage clustering or k-means [5]. By averaging the vectors of all anomalies in a cluster and pruning dimensions that occur in single or few instances only, we can construct a general vectorial model A for each attack cluster. The vector A comprises the average frequencies of prevalent feature strings in the considered attack cluster. We refer to the set of feature strings associated with A as a *unilateral signature*. Similarly, we can average and prune vectors of normal SIP messages to a vectorial model N. By computing $B = A - N$ we obtain a *bilateral signature*, B, that indicates malicious strings through large positive values and normal patterns by negative values [17]. Both types of signatures—unilateral and bilateral—can be represented as sets of strings and thus distributed to the majority of available signature-based systems. While unilateral signatures correspond to regular signatures as deployed in most systems, bilateral signatures improve attack detection by also incorporating traffic patterns characteristic for the particular IMS domain, such as typical SIP headers, whose absence also indicates anomalous activity.

The technique for embedding SIP messages to a vector space and measuring similarity of objects introduced in this section provides a generic in-

[2] Defensics is a trademark of Codenomicon, Inc.



(a) Global anomaly detection method

(b) Local anomaly detection method

Figure 5: Global and local detection performance.

strument for learning with SIP traffic, even beyond anomaly detection and signature generation [12]. For instance, embedded messages can also be processed using classification methods (e.g., for identification and categorization of unsolicited content such as spam over Internet telephony (SPIT)) and clustering algorithms (e.g., for identification of common groups of regular messages). Moreover, analysis of embedded messages may also help in identifying trends and deviations in SIP traffic as induced by temporal or local events, e.g., through learning methods for identification of principle and independent components in data [21].

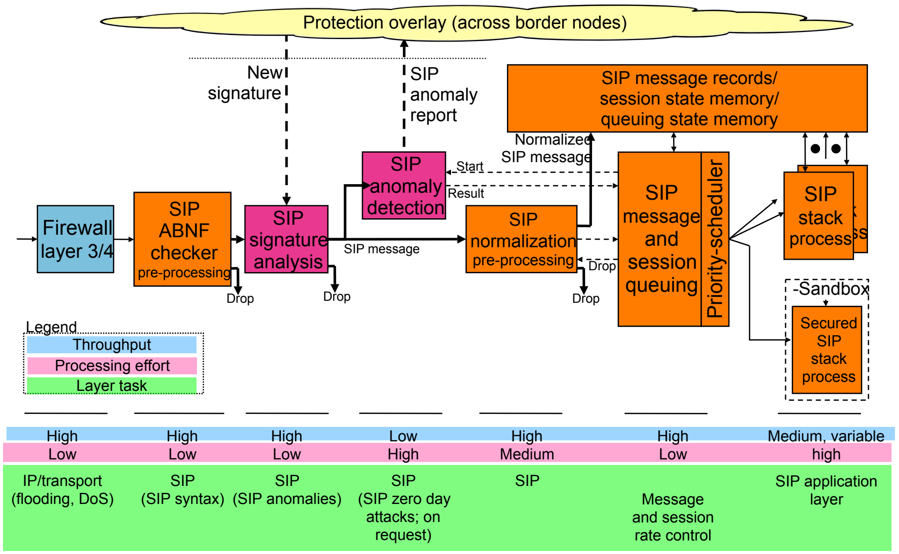
5 Advanced Architecture

We are now ready to present the advanced architecture of the SIP reactive security subsystem, illustrated in Figure 6. Two new and major components are introduced into the architecture: SIP signature analysis and SIP anomaly detection. Here, the SIP pre-processing module initially shown in Figure 3 is separated into two sub-modules. The SIP ABNF checker sub-module remains in its original position, whereas the SIP normalization sub-function processes the SIP messages after the SIP signature analysis and the anomaly detection have been performed. The SIP ABNF checker can be designed with very high throughput and is therefore inserted directly into the SIP message path. A SIP ABNF checker implementation by Bell Labs' Alan Jeffrey proofed this high efficiency (high throughput by low processing effort used). The theoretical background of the applied algorithm is explained in [2].

The SIP signature analysis module uses signatures to identify known anomalies within SIP message streams. This module also can be inserted into the data stream as it does not hinder the throughput. The key difference from previous approaches to SIP security is that the set of signatures can be updated with new signatures which are derived from the loop consisting of SIP anomaly detection and the protection overlay.

The SIP anomaly detection as explained in the previous section is not inserted directly into the SIP message data path due to a relatively high computational effort per SIP message, especially for local anomaly detection techniques. The session queue module decides for current session and/or client states which SIP message has to be analyzed by the anomaly detection module. Based on the results from this module, the session queue module can forward the message either to one of the SIP stacks, or to a secure SIP stack running in a sandbox, so that a malicious message does not disturb the operating throughput of the border node. If the SIP anomaly detection module identifies any message anomaly, it reports this anomaly towards the protection overlay where further offline processing can be performed.

The current SIP anomaly detection module processes single SIP messages and makes a decision on each individual SIP message without being aware of



High	High	High	Low	High	High	Medium, variable
Low	Low	Low	High	Medium	Low	high
IP/transport (flooding, DoS)	SIP (SIP syntax)	SIP (SIP anomalies)	SIP (SIP zero day attacks; on request)	SIP	Message and session rate control	SIP application layer

Figure 6: Advanced SIP security architecture. Abbreviations: ABNF–Augmented Backus-Naur form, DoS–Denial of service, IP–Internet Protocol, SIP–Session Initiation Protcol.

any sessions, clients, or history in general. But it may happen that application layer attacks may not be visible by investigating single SIP messages or a sequence of SIP messages without being aware of any states or history. The applied machine learning algorithm used by the SIP anomaly detection module allows the module to be "trained" by using enhanced SIP messages where additional proprietary header fields for session context are appended, as shown in Figure 7. These fields may contain supplementary session state information or more generic and higher layer information on client, node, or domain states. The anomaly detection inserts the additional header fields generated using a context database referenced by a client or session identifier (ID) from the original SIP message. The resulting extended SIP message is not standards-compliant, but the anomaly detection module was trained with these extended messages. The successive module receives the decision of the anomaly detection module and adds these results to the context database, which may in turn effect the decisions of the messages to come.

6 Domain-Wide Protection Solution

One of the important features of the advanced security architecture of a border node is the interface to the protection overlay. The protection overlay provides additional non-real time functionality for hardening the security of border nodes. This functionality also contributes to acceleration of processing on a SIP path because any occurrence of a zero-day attack(s) at any border node can immediately be taken into account without a need to resort to a relatively complex anomaly detection unit. The main functionality of the protection overlay is to derive an appropriate signature from the anomaly reports (as explained earlier) and to distribute it as soon as possible to all border nodes connected to the protection overlay. Hereby, the protection overlay may span all border nodes belonging to the same domain, but in principle could even be expanded across border nodes of the same manufacturer deployed in different domains.

In order to ensure a high quality of generated signatures, various verification functions are carried out by the protection overlay, such as testing for false positives on clean data and incorporation of advanced diagnostic information from hardened SIP stacks. A general architecture of the domain-wide protection functionality is shown in Figure 8.

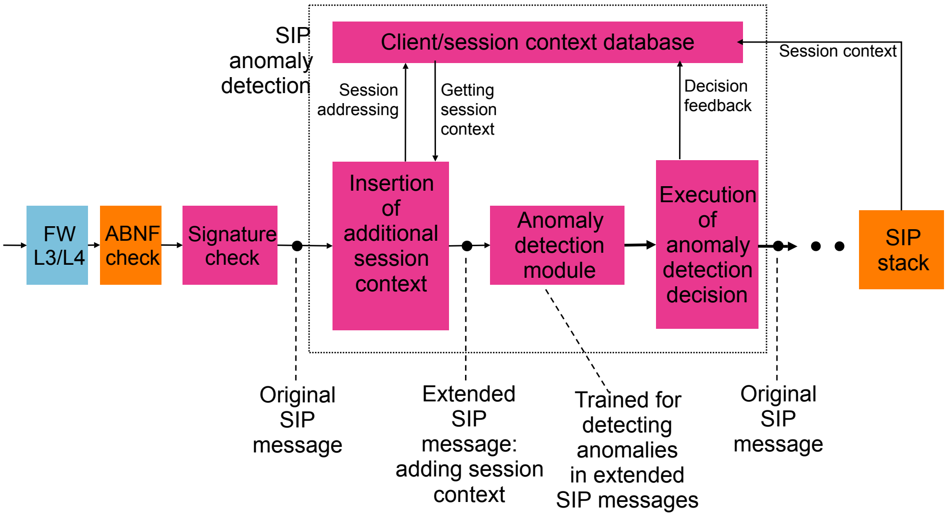

SIP
anomaly
detection
Client/session context database
Session context
Session addressing
Getting session context
Decision feedback
FW L3/L4
ABNF check
Signature check
Insertion of additional session context
Anomaly detection module
Execution of anomaly detection decision
SIP stack
Original SIP message
Extended SIP message: adding session context
Trained for detecting anomalies in extended SIP messages
Original SIP message

Figure 7: Extended anomaly detection by taking session context into account. Abbreviations: ABNF–Augmented Backus-Naur form, FW–Firewall, L3/L4–Layer 3/Layer 4, SIP–Session Initiation Protocol.

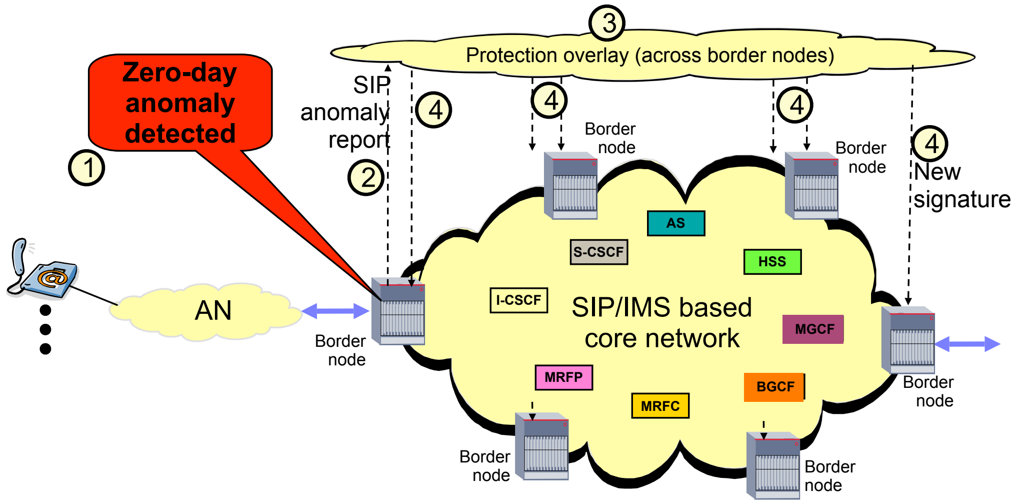


Figure 8: Protection Domain. Abbreviations: AN–Access network, AS–Application server, BGCF–Border gateway control function, HSS–Home subscriber server, I-CSCF–Interrogative CSCF, IMS–IP Mulimedia Subsystem, MGCF–Media gateway control function, MRFC–Media resource function controller, Media resource function processor, S-CSCF–Serving CSCF, SIP–Session Initiation Protocol.

7 Conclusion

The capability to detect previously unknown attacks is essential for the security of IMS infrastructures, especially in view of the long deployment horizon of such systems. The proposed architecture for enhanced protection of IMS edge nodes against novel attacks introduces an anomaly detection component that can effectively identify malicious anomalies in SIP traffic. The advanced architecture further extends this component with automatic signature generation that can be used for site-wide or domain-wide deployment. Experimental evaluation of the proposed architecture on real SIP traffic has verified excellent (up to 99 percent) detection accuracy for the system.

The principles underlying the proposed anomaly detection system—embedding of SIP messages in a high-dimensional features space and similarity-based operations—can be used for a number of other potential applications in the context of the IMS system. With the help of advanced machine-learning techniques, the proposed technology can be used for identifying clusters of related signaling patterns, finding unsolicited messages (SPIT), or performing advanced load balancing in the SIP stack. Future work will address the fine-grain incorporation of semantic information from all stages of SIP processing into the learning algorithms for further performance and accuracy improvements.

8 References

[1] H. Abdelnur, R. State, I. Chrisment, and C. Popi, "Assessing the Security of VoIP Services", Proc. 10th IFIP/IEEE Internat. Symposium on Integrated Network Management (IM '07) (Munich, Ger., 2007), pp. 373–382.

[2] M. Benedikt, A. Jeffrey, and R. Ley-Wild, "Stream Firewalling of XML Constraints", Proc. ACM SIGMOD Internat. Conf. on Management of Data (SIGMOD '08) (Vancouver, BC, Can., 2008), pp. 487–498.

[3] M. Cortes, J. R. Ensor, and J. O. Esteban, "On SIP Performance", Bell Labs Tech. J., 9:3 (2004), 155–172.

[4] T. Dagiuklas, D. Geneiatakis, G. Kambourakis, D. Sisalem, S. Ehlert, J. Fiedler, J. Markl, M. Rokos, O. Botron, J. Rodriguez, and J. Liu, General Reliability and Security Framework for VoIP Infrastructures, SNOCER Tech. Report D2.2, 2005, <http://www.snocer.org/>.

[5] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification, 2nd ed., John Wiley & Sons, New York, 2001.

[6] T. Joachims, Learning to Classify Text Using Support Vector Machines, Kluwer Academic Pub., Boston, MA, 2002.

[7] H. Kaplan and D. Wing, "The SIP Identity Baiting Attack", IETF Internet Draft, Feb. 2008, <http://tools.ietf.org/id/draft-kaplan-sip-baiting-attack-02.txt>.

[8] C. Krügel, T. Toth, and E. Kirda, "Service Specific Anomaly Detection for Network Intrusion Detection", Proc. ACM Symposium on Applied Comput. (SAC '02) (Madrid, Sp., 2002), pp. 201–208.

[9] P. Laskov, C. Gehl, S. Krüger, and K.-R. Müller, "Incremental Support Vector Learning: Analysis, Implementation and Applications", J. Mach. Learn. Res., 7 (Sept. 2006), 1909–1936.

[10] W. Lee, S. J. Stolfo, and K. W. Mok, "A Data Mining Framework for Building Intrusion Detection Models", Proc. IEEE Symposium on Security and Privacy (S&P âĂŹ99) (Oakland, CA, 1999), pp. 120–132.

[11] M. V. Mahoney, "Network Traffic Anomaly Detection Based on Packet Bytes", Proc. ACM Symposium on Applied Comput. (SAC '03) (Melbourne, FL, 2003), pp. 346–350.

[12] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An Introduction to Kernel-Based Learning Algorithms", IEEE Trans. Neural Networks, 12:2 (2001), 181–201.

[13] M. Nassar, R. State, and O. Festor, "Intrusion Detection Mechanisms for VoIP Applications", Proc. 3rd Annual VoIP Security Workshop (VSW '06) (Berlin, Ger., 2006).

[14] K. Oberle, S. Wahl, T. Strauss, S. Braun, and C. Pena, Flexible Multi-Service Edge Router, MUSE Report D B1.10 v01, Dec. 21, 2006,

[15] M. Poikselkä, G. Mayer, H. Khartabil, and A. Niemi, The IMS: IP Multimedia Concepts and Services, John Wiley & Sons, Chichester, Eng., Hoboken, NJ, 2006.

[16] K. Rieck and P. Laskov, "Detecting Unknown Network Attacks Using Language Models", Proc. 3rd Conf. on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA '06) (Berlin, Ger., 2006), pp. 74–90.

[17] K. Rieck and P. Laskov, "Language Models for Detection of Unknown Attacks in Network Traffic", J. Comput. Virology, 2:4 (2007), 243–256.

[18] K. Rieck and P. Laskov, "Linear-Time Computation of Similarity Measures for Sequential Data", J. Mach. Learn. Res., 9 (Jan. 2008), 23–48.

[19] J. Rosenberg and C. Jennings, "The Session Initiation Protocol (SIP) and Spam", IETF RFC 5039, Jan. 2008, <http://www.ietf.org/rfc/rfc5039.txt>.

[20] G. Salton, A. Wong, and C. S. Yang: "A Vector Space Model for Automatic Indexing", Commun. ACM, 18:11 (1975), 613–620.

[21] B. Schölkopf, A. Smola, and K.-R. Müller, “Nonlinear Component Analysis as a Kernel Eigenvalue Problem”, Neural Computation, 10:5 (1998), 1299–1319.

[22] M. Sher, Secure Service Provisioning (SSP) Framework for IP Multimedia Subsystem (IMS), Ph.D. Dissertation, Technical University Berlin, Dec. 2007.

[23] M. Siebert, B. Xu, M. Grigat, E. Weiss, N. Bayer, D. Sivchenko, T. R. Banniza, K. Wünstel, S. Wahl, R. Sigle, R. Keller, A. Dekorsy, M. Bauer, M. Söllner, J. Eichinger, C. Fan, F. Pittmann, R. Kühne, M. Schläger, I. Baumgart, R. Bless, and S. Stefanov, “ScaleNet – Converged Networks of the Future”, IT – Inform. Technol., 48:5 (2006), 253–263.

[24] D. Sisalem, S. Ehlert, D. Geneiatakis, G. Kambourakis, T. Dagiuklas, J. Markl, M. Rolos, O. Botron, J. Rodriguez, and J. Liu, Towards a Secure and Reliable VoIP Infrastructure, SNOCER Tech. Report D2.1, May 2005, <http://www.snocer.org/>.

[25] D. M. J. Tax and R. P. W. Duin, “Support Vector Domain Description”, Pattern Recognition Lett., 20:11-13 (1999), 1191–1199.

[26] K. Wang, J. Parekh, and S. Stolfo, “Anagram: A Content Anomaly Detector Resistant to Mimicry Attack”, Proc. 9th Internat. Symposium on Recent Advances in Intrusion Detection (RAID ’06) (Hamburg, Ger., 2006), pp. 226–248.

[27] S. Wanke, M. Scharf, S. Kiesel, and S. Wahl, "Measurement of the SIP Parsing Performance in the SIP Express Router," Proc. 13th Open Eur. Summer School and IFIP TC6.6 Workshop (EUNICE ’07) (Enschede, Neth., 2007), published in Dependable and Adaptable Networks and Services, Lecture Notes in Comput. Sci. (LNCS 4606) (A. Pras and M. van Sinderen, eds.), Springer, Berlin, Heidelberg, New York, 2007, pp. 103–110.