A Comprehensive Framework for Detecting and Preventing VoIP Fraud and Misuse

Dirk Hoffstadt, Erwin Rathgeb
Computer Networking Technology Group
University of Duisburg-Essen
Essen, Germany
{dirk.hoffstadt,erwin.rathgeb}@iem.uni-due.de

Matthias Liebig
ISACO GmbH
Berlin, Germany
matthias.liebig@isaco.de

Ralf Meister
genua mbH
Kirchheim, Germany
ralf_meister@genua.de

Yacine Rebahi, Tran Quang Thanh
Fokus Fraunhofer
Berlin, Germany
yacine.rebahi@fokus.fraunhofer.de

***Abstract*—The SUNSHINE framework presented in this paper aims at detecting and preventing VoIP fraud and misuse. The SUNSHINE architecture is modular, and considers both prevention and detection at the network as well as the application level. SUNSHINE offers a combination of firewalling and intrusion detection, a distributed sensing system, a CDR analysis based on statistical analysis and artificial intelligence, a component for correlating and aggregating alarms, and a DNS-based real-time blacklist for VoIP. The SUNSHINE framework has been implemented based on components contributed by the project partners and first insights from initial deployment are already available.**

Keywords—VoIP, SIP, fraud, misuse, CDR, firewall, sensor, NN-SOM, profiling, eRBL, SUNSHINE

I. INTRODUCTION

Voice-over-IP (VoIP) communication based on the Session Initiation Protocol (SIP) [1] has evolved as de-facto standard for voice communication and, support of open IP-based interfaces is increasingly important. VoIP is subject to the fraud schemes known from traditional telephony services as well as those known from today's Internet as VoIP blends these technologies. In addition, VoIP opens up new opportunities for misuse and fraud. SIP servers allowing access from external networks are subject to fraudulent registration attempts and subsequent calls via compromised SIP accounts. This is attractive for attackers, because they can gain immediate financial benefit by making toll calls which are charged to the account of the victim potentially causing substantial financial damage in a very short time.

Legitimate users also have more opportunities to violate policies associated with their subscription, e.g. by—potentially globally—sharing their flat rate services. Such violations are difficult to detect and in-depth statistical evaluation of usage patterns is necessary to do so. While countermeasures for specific fraud and misuse schemes have already been proposed, a comprehensive framework protecting users as well as service providers does not yet exist. This paper proposes such a framework which uses inputs from various sources and related to different protocols and layers, aggregates them and provides the information about detected attacks and violations to the relevant devices, e.g. firewalls or SIP servers, in a flexible and unified way.

This paper is organized as follows. The second section includes an overview of VoIP, taxonomy of VoIP fraud and misuse, and a brief state of the art discussion. The third section discusses the overall SUNSHINE architecture and Section IV gives details on some of the major components.

II. BACKGROUND

A. Session Initiation Protocol (SIP)

SIP is an application-layer control protocol that allows users to create, modify and terminate sessions with one or more participants. It can be used to create two-party, multiparty, or multicast sessions that include Internet telephone calls, multimedia distribution, and multimedia conferences.

In SIP, a user is identified by a SIP URI in the form of user@domain. This address can be resolved to a SIP proxy that is responsible for the user's domain. To identify the actual location of the user in terms of an IP address, the user needs to register his IP address at the SIP registrar responsible for his domain. Thereby, when inviting a user, the caller sends his invitation to the SIP proxy responsible for the user's domain, which checks the location of the user in the registrar's database and forwards the invitation to the callee. The session initiation is then finalized by a SIP message exchange between caller and callee to negotiate the addresses at which they would like to receive the media and what kind of media they can accept. After finishing the session establishment, the end systems can exchange data directly without the involvement of the SIP proxy.

B. Fraud and Misuse in SIP-based Networks

Different definitions of fraud and misuse are reflected in the literature. In general, fraud can simply be seen as any activity that leads to the obtaining of financial advantage or causing of loss by implicit or explicit deception. We distinguish between a third party misuse in SIP-based networks which can be

detected in real-time by observing the SIP signaling messages and other related traffic and fraud by legitimate service users that can be detected by using offline analyses based on usage records (Call Data Record, CDR).

Examples for the first type of misuse are:

- Unauthorized registration at a third party SIP account (Registration Hijacking) and subsequent unauthorized calls (Toll Fraud). Preparation of Toll Fraud requires to first detect SIP servers (Server Scan) and existing accounts (Extension Scan). All stages show specific SIP message patterns [14] allowing real-time detection.
- Attack on the configuration of a VoIP telephone with IP methods (e.g. via Web interface) and modification of settings to forward incoming calls to value-added services the offender benefits from. This requires cross-protocol monitoring for detection.
- Use of media channels negotiated during the session initiation to open ports on firewall systems. These can be used to elude the security policy of the firewall by misusing the channels for other non-media uses.

Examples for fraud by legitimate users include:

- Abuse of flat-rate services intended for personal use only. Some subscribers offer this service to other people resulting in high usage and losses to the operator.
- Service usage not matching the subscription type, e.g. using a residential service, which is usually cheaper, for business purposes. Another example is where the customer subscribes for the option that allows him to use its own PBX, and then use this PBX as a dialer for call center purposes. This may severely affect the performance of the VoIP provider's platform.

C. Related Work

SIP is IP-based, which means that all the threats known in the IP environment can be inherited by VoIP. Denial-of-Service (DoS) attacks, spoofing and man-in-the-middle attacks are examples of threats that VoIP has inherited from the IP infrastructure. In addition to that, there exist other threats that are SIP-specific, such as, registration/call hijacking, impersonating a SIP server, tampering with message bodies, abnormally setting up and tearing down sessions, and SPIT (Spam over Internet Telephony).

To make SIP secure, some standard solutions were suggested. SIP supports hop-by-hop security using Transport Layer Security [1] and end-to-end security using Secure MIME (S/MIME) [1]. However, these solutions cannot be effective against the fraud and misuse problem.

a) IP-based Detection Methods

A detailed up-to-date analysis of VoIP attacks against Honeypots is given by Valli [14]. The source data is captured at a Honeypot system consisting of several virtualized Low Interaction Honeypots that are logging to the same system. A simple statistical analysis is performed. In [9] an initial SIP-based Honeypot System and an analysis of SIP attack traffic are presented. This analysis is the basis for signature-based misuse detection, because it is important to understand the attacker's behavior first. Furthermore, in [13] the SIP Trace Recorder (STR) is presented, which allows passive attack monitoring in SIP-based networks.

b) Data Mining for CDR Analysis

A multi-dimensional approach to fraud prevention is needed that will incorporate SIP security as well as data mining. In data mining, both supervised and unsupervised learning are used. In the former, extensive training using labeled data of both fraudulent and non-fraudulent cases is performed. This training establishes a model which allows classifying new cases as legitimate or fraudulent. The use of supervised techniques was not an option for us due to the difficulty of obtaining training data.

The unsupervised learning category is being used when there is no a priori labeling. Profiling, clustering, and Neural Network Self Organizing Map (NN-SOM) are interesting examples of this category that were implemented in our work. Issues such as profiles implementation/update, data fluctuation, and metrics used for misbehavior detection [7], [8] were also investigated in the design process.

Fraud detection solutions were mainly developed by companies to protect their assets or as commercial products and were not disclosed, except few related papers that were based on the use of artificial intelligence [19].

c) Prevention Methods for IP Networks

Misuse and fraud are no new issues in IP networks. For protection in IP networks one can distinguish between active and passive methods. Active methods are typically implemented by using firewalls. Firewalls separate the network into segments, and control which connection may pass the firewall. Depending on the input used for the decision, there is a distinction between packet filters that use the content of the IP headers, and the so-called application layer gateways (ALG) that also include the payload. For a description of firewall types with application cases and pros and cons of the two firewall types see e.g. [11].

The passive methods for network security include intrusion detection systems (IDS). These systems monitor the network and search for suspicious traffic. Once suspicious traffic is detected an alarm is generated. Any reaction triggered by this alarm depends on the application. For application cases of IDS see e.g. [12].

III. THE SUNSHINE APPROACH

A. Objectives

The framework has been developed in and named after the SUNSHINE project [3] funded by the German Federal Ministry of Education and Research (BMBF). It has been developed to detect and prevent fraud and misuse in VoIP environments and has the following characteristics,

It combines real-time detection and offline statistical analysis components. This allows exploiting synergies between both approaches which have been used independently in the past.

It uses different data sources. It uses on one hand input data generated by online monitoring of relevant network traffic, in particular SIP signaling messages, and on the other hand

offline data collected in CDRs. The inputs are aggregated and correlated to achieve better detection accuracy.

The suggested solution is multi-layered. It uses different algorithms and techniques including, rules, profiling, Neural Networks, and clustering.

It offers monitoring and intervention options. The result of the analysis can either be used by a passive monitoring solution, or to trigger actions, e.g. in firewalls or SIP servers.

It offers a central fraud information system. Similar to DNS real-time blacklisting, this system is fed by various detection modules and the information collected can be shared by these modules as well other components distributed in the Internet which extends the detection outreach significantly.

B. High Level Architecture

Fig. 1 depicts the main functional blocks of the SUNSHINE framework.

a) Input Layer

The top of the figure shows the input data layer. SIP messages are transported in IP traffic which can also carry configuration commands for SIP devices (e.g. via HTTP). Therefore, IP traffic is one of the input sources. SIP servers process and store call-related data in form of CDRs which are collected by the VoIP providers for billing purposes. This is the second major input source. CDRs can be generated from observed SIP traffic. However, CDRs only retain aggregated information related to (successful) calls and do not allow detection of SIP-based attacks as, e.g., SIP server scans.

b) Analysis Layer

The detection of fraud and misuse is performed in several analysis components to account for the diversity of input data. The Distributed Sensor System provides online detection of SIP-specific attacks. It operates rule-based and provides stateful analysis of relevant network traffic. The BRO traffic analysis (an extension to the *bro* network security monitor [6]) also provides online traffic analysis and is tightly coupled to the firewall component. The CDR analysis component provides offline processing of CDRs by means of data mining methods.

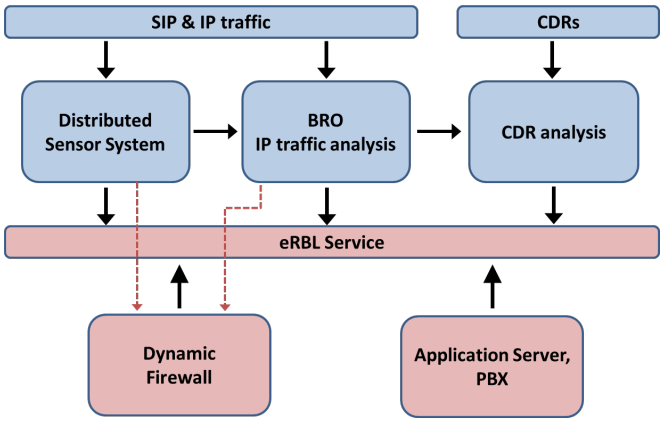

Fig. 1. SUNSHINE framework architecture

c) Communication Layer

The results of the analysis step can either be used for alarm generation or else be used by active components for automatic call intervention. To provide the analysis results in a way suitable for the active components a specific service, the eRBL (extended Real-time Blacklist) has been defined. It is fed with analysis results and can be queried by the active components in a unified way.

In addition, there is a private communication channel between the BRO module and the firewall which allows direct, real-time interaction. Furthermore, the BRO module and the firewall interface can be directly triggered by the Distributed Sensor System to react to attacks in real-time.

d) Active Components

The bottom layer in Fig. 1 represents the active components. Both, the VoIP provider and the (enterprise) customers can benefit from the SUNSHINE framework. Thus the application server components at the VoIP provider side and the Internet firewall at the (enterprise) customer side feature specific communication interfaces to the SUNSHINE communication layer.

C. Benefits of the SUNSHINE Architecture

Interoperability: The SUNSHINE framework uses existing and proven methods for its basic components. This helps it to easily interoperate with a wide range of products and platforms. Technologies such as DNS, BRO, DIAMETER (RFC 3588) are being used.

Distrubuted detection and central information sharing: SUNSHINE uses different detection and prevention layers including spatially distributed sensors. It takes spatially distributed network traffic and CDRs as input for analysis and adds a central communication layer allowing SUNSHINE modules and active components to share information.

Cross-layer detection: Contrary to the traditional setup, both methods of network security—firewalls and IDS—are combined to strengthen their effects. The tight coupling of firewall and IDS makes it possible to incorporate the IDS's global view of network activities into the decision of the firewall if a connection may pass.

Benefits both operators/VoIP providers as well as enterprises (customers). The services offered by the SUNSHINE framework can be used by both, VoIP customers and VoIP providers.

IV. Main SUNSHINE Components

A. CDR Analysis

The CDR analysis has a modular architecture. This permits the incorporation of additional detection, correlation, analysis, and notification tools. Some of the detection algorithms need to be scheduled over sufficiently large time intervals to be able to operate. The profiling-based technique is a particular case of such techniques. In contrast to this, a rule-based technique can be launched on demand. Indeed, the rule engine can be configured to apply a given rule to any new call (or CDR) that is made to suspected destinations. In addition to that, an alarm

can be sent (in urgent cases) by e-mail or by another means to the fraud management expert. For these reasons, we decided to implement the CDR analysis framework in an event-based manner, i.e., the components communicate by generating and receiving notifications. An event reflects the occurrence of an item of interest to some of the system components, e.g., the arrival of a new CDR or the creation of a new rule. The event-based architecture is well suited for large scale distributed applications and provides easy integration of autonomous and heterogeneous components. The CDR analysis part is composed from the following components as shown in Fig. 2:

- A fraud management interface.
- The detection techniques and algorithms. The techniques include a rule engine, call profiling, geolocation profiling, Top-k, and Neural Networks Self Organizing Map (NN-SOM). In this paper, only the profiling techniques will be briefly discussed due to space constraints.
- The event-based system. The latter is the backbone that coordinates the tasks and links the different components together. This system is based on the XMPP protocol.

Profiling is an unsupervised technique which can be used to distinguish between legitimate and fraudulent subscribers' activities. The past behavior of the user is cumulated to build a profile that will be used to predict the user's future behavior [7], [8].

a) Call Profiling

The goal of the call profile is to use the activities related to legitimate calls as a basis for fraud detection. This kind of profiling is designed in the following way,

- Short-term (hour/daily basis) and long-term (weekly/monthly basis) profiles based on features such as number of premium/international calls and call durations are built. The long term profiles are statistical summaries using trimmed mean, Median Absolute Deviation, or moving average.
- Data fluctuation in the service usage is an important issue when building the profiles because activities vary from one day to another and periods of inactivity regularly occur. To cope with this challenge, the day is divided into four time slots (morning, afternoon, evening, night). The inactivity periods (reflected by the 0 values) are not taken into account when computing the long-term profiles since fraud is strongly bound to service usage.
- Detection of misbehavior by comparing the long-term and short-term profiles using an appropriate metric. In our implementation, z-score, Median Absolute Deviation, and Hellinger distance were implemented and tested.

b) Location Profiling

For each subscriber, a location profile is built based on the IP address of the caller typically included in the CDR. A geolocation database allows to determine where the service users are physically located based on their IP addresses. This can be used in an effective way to detect potential fraud by comparing the subscription location and the current user location. For the geolocation database, *geolitecity* from MaxMind [2] is used in our current solution. When the input

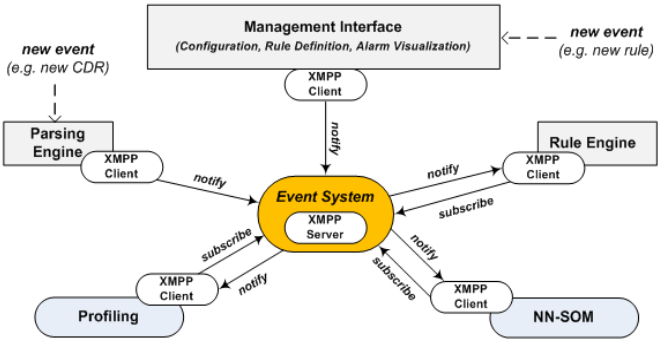

Fig. 2. CDR analysis architeture

IP address is, e.g., 217.159.49.6, *geolitecity* will output location number 57 (which belongs to Germany). Another important feature that MaxMind offers is the incorporation of open proxy detection. As fraudsters often use proxies to hide their identities, integrating mechanisms to detect such components proved to be crucial.

The location profile is built as follows: The first time the user starts using the service, the geolocation of the used IP address is determined and stored together with a timestamp. If over time another IP address was used by the caller, its corresponding geolocation information is not stored except if it differs from the previous one. In this case, a timestamp reflecting the last time the previous location was seen is also stored. The location profile can be used in different ways. In fact, it is impossible for a subscriber to make calls from different places (e.g. different countries) in a short period of time. If this occurs, this means that the same account is being used by two different persons—which might be a fraud indicator. This suspicion is reinforced if one of the persons using the account is behind an open proxy. Based on the user location profile, we calculate the distance between two consecutive calls made from different locations by using the "spherical law of cosines" formula. The location profile can also be used by the rule-based system to check whether the location change occurred in a country that is blacklisted which gives a stronger indicator of fraud. It can also be used as input data to more complex techniques such as NN-SOM. One of the main advantages of location profiling is its ability to operate in a near real-time manner compared to call profiling and NN-SOM.

B. Distributed Sensor System

Our analyses of the SIP traffic in a Honeynet [9] have revealed that attacks in SIP-based networks show specific message patterns that can be used for detection. We developed a system for distributed signature-based misuse detection, which consists of two parts: The passive light-weight monitoring Security Sensor on different devices and the Sensor Central Service (SCS) server. The traffic analysis can be performed in real-time. Fig. 3 shows that an interface (SSI) between Sensor and SCS is used for configuration, signature distribution and report reception.

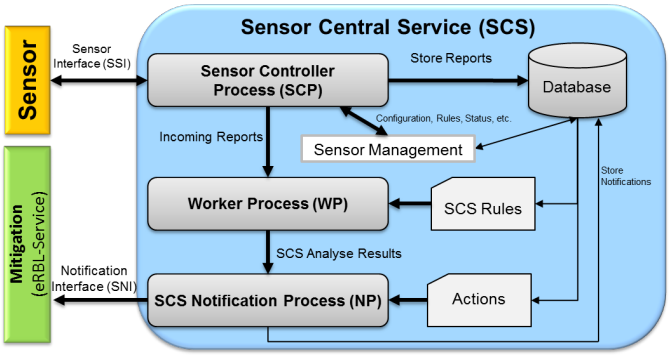

Fig. 3. Distributed Sensor System overview

a) Security Sensor

The Sensor component is a software tool for signature-based detection and reporting of misuse in SIP-based networks on different platforms and devices. It recognizes sequences of SIP messages that are defined in XML signatures and can report information about recognized message sequences, their source and their destination to the SCS. We implemented the Sensor component in C++ using *libpcap* [4] for easy access to network interfaces, filtering of network traffic and platform independency. The process of misuse detection and reporting in the Sensor executes three different steps which are performed by specific modules: The Listener module captures all SIP messages from a network interface in promiscuous mode and enqueues them in a FIFO queue. The Analyzer module accesses the queue and analyses the messages by using the pre-defined signatures.

Every rule specifies a sequence of SIP messages. During an Extension Scan for example, an attacker sends multiple REGISTER requests to a SIP server to detect active SIP accounts. A corresponding signature would scan for a sequence of e. g., three REGISTER requests from the same source IP to the same destination IP, but directed to different extensions. In addition, we also exploit timing information, because automated attacks result in multiple requests per second. Hence, setting the timing condition for this signature to 3 messages per second filters out manual attempts. The Analyzer compares every received SIP message to the first message of every signature also taking into account the timing condition. If a received message matches this first message, the signature state is copied and updated so that messages received later are compared to the next message of the signature. One message can lead to the update of more than one signature state. If the inspection of the messages fails, the signature is not updated. If a message matches no message of a signature it is discarded, else it is stored for comparison with messages received later.

If an attack is successfully detected (signature matched), the Notification module triggers SCS by using the SSI. For each signature a report with configurable parameters can be defined. It contains at least information about the source (IP address and port) and destination of an attack, the ID of the signature relating to the attack and the timestamp of the attack. Furthermore, the Sensor provides extended reports to analyze further SIP header values like the user agent.

b) Sensor Central Service (SCS)

The SCS consists of three modules as shown in Fig. 3. The Sensor Controller Process (SCP) allows easy distribution of configuration and control information to remote Sensors via the SSI. Configuration values and signatures can be deployed to different instances of the Sensor component, commands (e.g., restart, stop) can be sent and the attack reports of all Sensors can be collected via the SSI. The Sensor acts as a client only and communicates with the server using XML messages exchanged via persistent HTTPS connections. A HTTP POST request containing a well-defined XML structure is periodically sent by the Sensor. The server replies using HTTP status codes and another well-defined XML structure. The server identity is guaranteed by using an own Certificate Authority (CA). Furthermore, the Sensor authorizes itself by sending a sensor ID and a secret with every request. HTTP keep-alive allows to keep the connection persistent over a long time period.

The main module is the Worker Process (WP) which provides a rule-based analysis. In contrast to the signature-based detection at the Sensors, the WP uses more flexible event-based aggregation and detection rules based on PHP [10] scripts. The operator can define rules according to one or more incoming signature IDs or Sensor IDs. If a new Sensor report is received that corresponds to a defined SCS rule, the PHP script is executed. In addition to aggregation of incoming Sensor reports, the SCS can perform an eRBL query (see section IV.C) to request more detailed attacker information and alarm generation to inform mitigation components. If a SCS rule matches indicating a successfully detected attack, the Notification Process (NP) performs pre-defined, rule-specific notification actions automatically by using the eRBL REST interface. In case of activated real-time attack mitigation, the firewall is immediately triggered by the SCS to block further attack attempts.

We have installed two Sensors at different locations in Germany since November 2012 which send reports to our SCS server. We received over one million reports so far by using simple signatures for multi-stage Toll Fraud attacks. A first analysis shows that 11 source IP addresses were detected by both Sensors indicating that the aggregation of distributed Sensor information is actually beneficial.

C. Real-time Blacklists

Common instruments for fighting spam are so-called Real-time Blackhole Lists (RBL). Those lists provided by various operators [15] contain IP addresses suspected of spam or abuse. Mail servers use the Domain Name System (DNS) protocol to query the RBLs. In the lookup the client's IP address is given in reversed order. The DNS answer message determines whether the IP is on the blacklist and hence an A record is returned. The A record contains a loopback IP address that is used as a code e.g. for the reason of the listing [16]. In case the queried IP address is not blacklisted, the DNS answer indicates a name error.

Here, we introduce the so-called "Extended Real-time Blacklist" (eRBL) which adapts this mechanism to VoIP. Instead of e-mail messages SIP calls are checked for their

blacklist status. The reasons for blacklisting are similar: suspicion of fraud, misuse, and Spam over Internet Telephony (SPIT). Querying the eRBL is also done via DNS. While conventional RBL lookups focus solely on the sender's IP address, the eRBL interface allows querying the SIP URIs and telephone numbers of the caller and the callee as well as the source and destination IP addresses. Furthermore, it is also possible to combine any of those parameters in a single query to determine the blacklist state of the entire call rather than just of one of its participants. The result of an eRBL query is a DNS TXT record containing aggregated alarms from the analysis systems presented earlier. There is also a grading system for the alarms indicating the level of suspicion in percent. The rating is given in the TXT record for each alarm and also as a total value. Overall, a generic format for the answer message is used that supports further detection systems and methods. For the e-mail blacklists submission of entries differs depending on the list operator. For eRBL we specified an interface for creation and modification of entries based on the Representational State Transfer (REST) [17], as shown in Fig. 4. This interface uses HTTP requests for transfer and the JavaScript Object Notation (JSON) format [18] for representing the data. Users of this interface are the analysis systems (automatic submission) and also administrators, which may manually adjust, remove, or add entries.

There are many possible applications for the eRBL. A lookup can be added to firewalls or Session Border Controllers (SBCs) at the provider's side protecting the platform from attacks or known fraudsters. A VoIP provider who is under attack would submit information about it to a globally available eRBL. Another provider can benefit from this entry and either block the corresponding IP addresses proactively or always use the eRBL lookup upon reception of SIP requests to decide whether to accept them. Using the eRBL is not limited to providers, a PBX host within an organization or company could also profit from it. Also, the exchanged information may cover fraud and SPIT. The combination of multiple eRBL servers is possible via DNS zones—allowing a globally distributed infrastructure—and via the REST interface.

When dealing with fraud or registration hijacking, generated alarms are almost always treated as warnings or hints. There is rarely certainty whether a user's behavior is fraudulent or just non-ordinary. The eRBL entry should reflect this by having a relatively low initial rating, e.g. 40%. Now, when this suspect makes another call, it might be useful to verify the user's identity. This can be achieved with a SIP application server at the provider's platform or by extending the PBX. After the eRBL lookup, the user would be asked for a separate PIN only the account's owner would know before establishing the call to the dialed number.

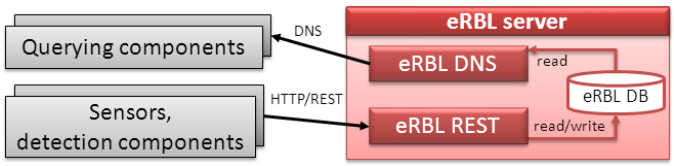

Fig. 4. Blacklist submissions and queries

Another possibility is to play an announcement stating the suspicion that the VoIP account might be compromised and the password should be changed.

D. *Intervention with Firewall and Intrusion Detection Systems*

Firewall systems offer protection against attacks that target weaknesses at the network and application layer. The protection is active in the sense that once a misuse is detected an immediate reaction can be performed. The analysis of misuse is based on the traffic payload. Therefore, a firewall in the form of an application level gateway (ALG) is appropriate, which requires a proxy process for forwarding data to both sides of the connection.

a) Combination of Firewall with IDS

The detection of misuse is based on data that is available to the firewall. Usually, the data consists of the payload of a single TCP or UDP session alone, and does not involve data of other sessions. Contrary to firewall systems, intrusion detection systems can monitor the entire network and thus have a broader scope than just one session. To broaden the analysis possibilities of firewalls, a combination of firewall and intrusion detection system is, therefore, beneficial. The intended combination of a firewall and intrusion detection system requires both extensions on the firewall and the IDS side.

b) Components

The firewall is based on *OpenBSD* [5], in which the focus of development is on security. *OpenBSD* includes a general purpose proxy named *relayd*. *relayd* is a user level program that uses the socket interface to handle network data. The socket interface is located at layer 4 of the OSI reference model. Therefore, the data processed by *relayd* is also located at layer 4.

The IDS chosen is *bro* [6]. *bro*'s analysis ranges from layer 2 to layer 7, supports a large number of network and application protocols, and offers the ability to add and customize so-called policy scripts for analysis. Contrary to a firewall, an IDS monitors the entire network. *bro* uses the *libpcap* [4] interface for data input, which is located at layer 2 of the OSI reference model. Since firewall and IDS work at different layers, an adaptation is necessary. The decision was to extend *bro* to allow input of data at layer 4.

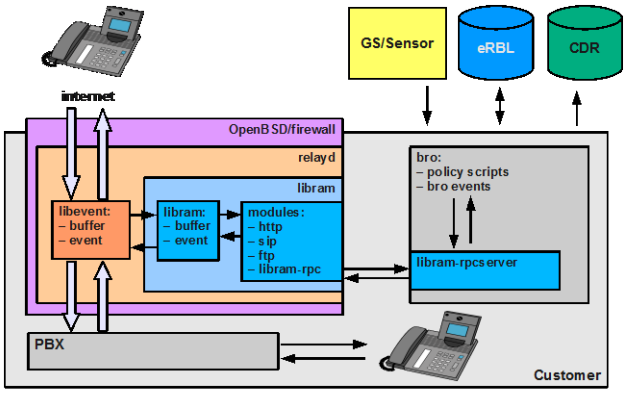

Fig. 5. Combination of firewall and intrusion detection system (*bro*)

c) Communication Protocol *libram*

Passing the data from the firewall to the IDS requires a protocol. Whereas at layer 2 one usually uses data in *libpcap* format even for transport purposes, no such library exists at layer 4. Therefore, a layer 4 library with the name *libram*—remote analysis and modification—was established. The design of *libram* involves several issues.

- It is used for passing data at layer 4. There is a client and a server part. The client part is integrated into *relayd* whereas the server part is integrated in *bro*. By connecting several *relayd* client instances to one *bro* server instance, this allows the single instance of *bro* to analyze the data from all connected *relayd* instances
- The usage of *libram* is not limited for passing data. Instead it offers an interface (in the form of an application programming interface) by which analysis modules can be plugged into *relayd*. The modules are implemented in the form of shared libraries that are embedded in the proxy process.
- One of these analysis modules is the rpc (remote procedure call) module that offers the facility of passing data for remote analysis. By using rpc methods, the application programming interface is extended to a network protocol that may be used by other clients.
- Other modules may not need a separate analysis process and may be executed within the proxy process itself.

The extensions of *relayd* and *bro* consist in the integration of the client respectively the server part of the communication library *libram* to *relayd* and *bro*.

d) Additional Input and Output

The SUNSHINE framework offers additional data from the eRBL and the sensor system. To include this additional input in the analysis process minor extensions to the existing methods in *bro* are necessary. In the case of eRBL the resolver part of *bro* had to be extended to allow queries of TXT records. For passing data from the sensors the communication library *broccoli* is used, which is part of the *bro* distribution. *bro* policy scripts are used to generate CDRs and eRBL records, and to push this data to the corresponding servers.

V. CONCLUSION

In this paper, we introduced the SUNSHINE framework architecture and its implementation. The scope of SUNSHINE was the development of a flexible framework capable of detecting fraudulent activities in VoIP networks. It offers a multi-layered solution to deal with fraud and service misuse in VoIP networks. The first line of defense is a distributed sensor system that scans for attack patterns derived from real-life attack. The second line of defense is based on Deep Packet Inspection (DPI) combining firewalling and intrusion detection systems. The third line of defense is achieved at the CDR level where the related data is analyzed using a mixture of expert systems and unsupervised learning. All the SUNSHINE framework components are fully implemented and are currently being integrated and tested. Initial field tests with the Distributed Sensor System and evaluations of the CDR analysis components clearly indicate that the combination and aggregation of heterogeneous inputs and analysis methods provides improved detection accuracy.

REFERENCES

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, "SIP: Session Initiation Protocol," RFC 3261, June 2002.

[2] MaxMind, www.maxmind.com, 14.03.2013.

[3] The SUNSHINE project, www.sunshineproject.net, 14.02.2013.

[4] Tcpdump & libpcap, www.tcpdump.org, 20.02.2013.

[5] OpenBSD, www.openbsd.org, 20.02.2013.

[6] The bro network security monitor, www.bro-ids.org, 20.02.2013.

[7] C.S. Hilas and J.N. Sahalos, "User profiling for fraud detection in telecommunication neworks," in *5th International Conference on Technology and Automation*, Thessaloniki, Greece, October 2005, pp 382-387, icta05.teithe.gr/papers/69.pdf, 14.03.2013.

[8] C. Cortes and D. Pregibon, "Signature based methods for data streams," Data Mining and Knowledge Discovery Journal, Springer, vol. 5, 2001, pp. 167-182.

[9] D. Hoffstadt, A. Marold, and E.P. Rathgeb, "Analysis of SIP-based threats using a VoIP Honeynet System," in *Conference proceedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications* (TrustCom-2012), Liverpool, UK, 2012.

[10] PHP Hypertext Preprocessor, www.php.net, 04.03.2013.

[11] K. Scarfone and P. Hoffman, "Guidelines on firewalls and firewall policy," NIST: National Institute of Standards and Technology, Special Publication 800-41, Revision 1, http://csrc.nist.gov/publications/nistpubs/800-41-Rev1/sp800-41-rev1.pdf, 28.03.2013.

[12] K. Scarfone and P. Mell, "Guide to intrusion detection and prevention systems (IDPS) (Draft)," NIST: National Institute of Standards and Technology, Special Publication 800-94, Revision 1, http://csrc.nist.gov/publications/drafts/800-94-rev1/draft_sp800-94-rev1.pdf, 28.03.2013.

[13] D. Hoffstadt, S. Monhof, and E.P. Rathgeb, "SIP Trace Recorder: monitor and analysis tool for threats in SIP-based networks," in TRaffic Analysis and Classification Workshop (IWCMC2012-TRAC), Limassol, Cyprus, Aug. 2012.

[14] C. Valli, "An analysis of malfeasant activity directed at a VoIP Honeypot," in *Proceedings of the 8th Australian Digital Forensics Conference*, Perth, Australia, 2010, pp. 168-174.

[15] Spam Links, "DNS & RHS blackhole lists," spamlinks.net/filter-dnsbl-lists.htm, 14.03.2013.

[16] J. Levine, "DNS blacklists and whitelists," RFC 5782, Feb. 2010.

[17] R.T. Fielding, "Architectural styles and the design of network-based software architectures," UC Irvine, U.S., 2000.

[18] D. Crockford, "The application/json media type for JavaScript Object Notation (JSON)," RFC 4627, July 2006.

[19] H. Kvarnstrom, E. Lundin, E. Jonsson, "Combining fraud and intrusion detection—meeting new requirements—," in *Proceedings of the Fifth Nordic Workshop on Secure IT Systems* (NorSec2000), Reykjavik, Iceland, October 12-13, 2000.