Holistic VoIP Intrusion Detection and Prevention System

Mohamed Nassar, Saverio Niccolini, Radu State, Thilo Ewald

HAL Id: inria-00169036
https://hal.inria.fr/inria-00169036

Submitted on 31 Aug 2007

Holistic VoIP Intrusion Detection and Prevention System

Mohamed Nassar
LORIA - INRIA Lorraine
615, rue du jardin botanique
54602 Villers-Lès-Nancy
France
Mohamed.Nassar@loria.fr

Saverio Niccolini
NEC Europe Ltd.,
Network Laboratories
Kurfuersten-Anlage 36
69115 Heidelberg
Germany
Saverio.Niccolini@netlab.nec.de

Radu State
LORIA - INRIA Lorraine
615, rue du jardin botanique
54602 Villers-Lès-Nancy
France
Radu.State@loria.fr

Thilo Ewald
NEC Europe Ltd.,
Network Laboratories
Kurfuersten-Anlage 36
69115 Heidelberg
Germany
Thilo.Ewald@netlab.nec.de

ABSTRACT

VoIP security is crucial for current and future networks and services. The rapid shift from a closed and confined telephony towards an all IP network supporting end to end VoIP services provides major challenges to the security plane. Faced with multiple attack vectors, new and comprehensive defensive security solutions for VoIP must emerge from the research community.

This paper describes a multilayer intrusion detection and prevention system architecture for VoIP infrastructures. The key components of the approach are based on a VoIP-specific honeypot and on an application layer event correlation engine. While each component alone can detect only a subset of VoIP-specific attacks, the two of them together can provide an effective defense for the many class of attacks. We show in this paper, how different and complementary conceptual approaches can jointly provide an in depth defense for VoIP architectures.

1. INTRODUCTION

Securing VoIP infrastructures constitutes one of the major challenges for both the operational and research communities because security by design was not a key component in the early phases of VoIP research and development. VoIP-specific security solutions are currently demanded by the market while the research and standardization are still trying hard to address the issues of securing and monitoring VoIP infrastructures. Over the past few years, different approaches emerged, anyway most of them only address the defense against a subset of potential attack vectors.

We propose in this paper a holistic approach to VoIP security intrusion detection and prevention. Our approach is based on a combined VoIP-specific honeypot and application layer monitoring scheme based on SIP. Such an approach is capable of detecting multiple types of attacks: The VoIP-specific honeypot is best suited for preventing social attacks like Spam over Internet Telephony (SPIT) and VoIP Phishing (Vishing) as well as other stealthy reconnaissance actions, while the SIP correlation engine is adapted to detect Denial of service and/or fraudulent usage. Our paper is structured as follows: Section 2 starts with on overview of major threats that must be addressed by current and future VoIP infrastructures. The global architecture proposed in this paper is described in section 3, while the next two sections (4 and 5) describe in detail the two main security components of this architecture: a VoIP honeypot and respectively an anomaly based intrusion detection engine. An overview of related work is given in section 6. The paper ends with conclusions and pointers to future work in section 7.

2. VOIP THREATS AND PROBLEMS

VoIP security threats constitute a superset of those faced by data networks. The main source of security threats comes from the fact that both the signaling and the control plane in VoIP are carried over the IP network. Therefore, VoIP infrastructure shares the same vulnerabilities as the data networks. In addition, VoIP-specific threats (both at signaling and data layer) do also represent major causes of concern. Among the most dangerous VoIP-specific attacks, identity theft, eavesdropping, fraudlent usage and social attacks (SPIT, Vishing) are becoming reality. Denial of service (DoS) attacks can be oriented against the VoIP infrastructure (servers, proxies, agents) and lead to the crippling and total shut down of a VoIP infrastructure. Viruses, worms and backdoors can allow the remote control of IP phones and VoIP proxies allowing more than just simple malicious purposes. Password cracking bots can be launched and assure a fraudulent usage, privacy violation and SPIT/Vishing operations. In addition, with VoIP growing rapidly, hackers

are becoming interested in extending and benefiting from this market. Designing a security solution for a distributed, multi-protocol and QoS-sensitive application as VoIP is a hard task. This paper addresses the issue of a global and holistic security solution capable to deal with the multiple threats faced by a VoIP infrastructure.

2.1 Interception and Modification Threats

In contrast to the difficulties encountered by the illegal interception in PSTN, a VoIP conversation reconstruction is possible using traffic captured and decoded. Free software like Vomit[1] (Voice over misconfigured internet telephones) are already existing for enterprise and convenient Cisco IP phones. Similar tools for SIP have also emerged [3] and do not represent major technological difficulties. The attacker is left with a huge choice of actions ranging from injecting/eavesdropping and stealing sensitive financial and commercial information.

2.2 Denial of Service and Toll fraud

Flooding attacks can target the signaling plane elements (e.g. proxy, gateway, etc.) with the objective to take them down and produce havoc in the VoIP network. This is very easy to do by either flooding the signaling plane with a large quantity of messages, malformed messages or device specific vulnerabilities. Abuse of service attacks have as objective the fraudulent usage of the VoIP services. Threats that can be cited are unauthorized or unaccountable resource utilization that exploits specific vulnerabilities if the identity management of VoIP infrastructures aimed at fraudulous usage and/or spoofing call identification. For a comprehensive overview on the VoIP security threats, please check the taxonomy developed at [13].

2.3 Social Threats

Social threats are attacks ranging from the generation of unsolicited communications -which are annoying and disturbing for the users- to more dangerous data stealing (Phishing) attacks. The threat is classified as social since the term "unsolicited" is strictly bound to user-specific preferences and this makes hard for system to identify this kind of attack. An example of this is a threat commonly referred to as Spam over Internet Telephony (SPIT) (similar to Spam in the email systems but delivered by mean of voice calls) which leverage on the cheap cost that VoIP has with respect of legacy phone systems (it is currently estimated that generating VoIP calls is three order of magnitude cheaper than generating PSTN calls). SPIT calls can be telemarketing calls used for guiding callees to a service deployed to sell products. A subtle variant of SPIT is the so-called Vishing (VoIP phishing) attack that aims either to make the callee dialing expensive numbers in order to get the promised prize or to collect personal data redirecting the users towards Interactive Voice Responder (IVR) pretending to be trusted. Most of these attacks are going to be generated by machines (bot-nets) programmed to do such a job. Unsolicited communications (like SPIT or Vishing) are, from a signaling point of view, technically correct transactions. It is not possible to distinguish from the INVITE (in the case of SIP) if such a transaction is SPIT or not. From a technical point of view the challenge is even more complicated since the

---

[1] http://vomit.xtdnet.nl/

content is not available to help in the detection until the phone rings (disturbing the user) and the callee answers the call, for this reason techniques available from email spam like text filtering are hardly reusable. Even if a transaction is identified as unsolicited it depends strongly on the legal country environment how to handle it.

3. ARCHITECTURE FOR INTRUSION DETECTION AND PREVENTION

VoIP architectures are distributed (proxy servers, gateways, application servers, terminals, etc.) and thus difficult to tackle by a centralized security approach. The architecture of an intrusion detection and prevention solution should be distributed, unless the network is designed in a dedicated way. The architecture presented in this paper takes into account the possibility of using a single entry point but for the sake of generality it proposes a distributed approach combining a VoIP-specific Honeypot and application layer monitoring scheme based on SIP. Such an approach is capable of detecting multiple types of attacks basing on the combination of two solutions.

Figure 1 depicts the general architecture where the VoIP-specific Honeypot (see section 4) domain is separated by the real infrastructure domain. The distributed application layer monitoring is achieved deploying the VoIP Security Event Correlation (SEC) (see section 5) on VoIP infrastructural elements (SIP Proxy Servers, Terminals, etc.) both in the real domain and in the honeypot one.

In order to accommodate the VoIP-specific Honeypot into the intrusion detection and prevention architecture an additional domain was introduced. The task of this domain is to transparently bridge the internal domains (real and honeypot) and the external one correctly routing attacks to the honeypot domain and real requests to the real domain while hiding domain-specific information. In the architecture depicted in Figure 1 the domain is represented by an inbound / outbound proxy in a public reachable network.

The production domain (real) and the honeypot one are strictly disconnected from each other to prevent backdoors to the production environment when the VoIP-specific Honeypot is compromised. Thus a sharing information database and a black-listed users one are deployed in the inbound / outbound domain as information sharing mean between internal domains both domains. The scope of such databases is to have a sharing of information between the Honeypot domain and the real one to improve detection and prevention methods and schemes based on observations of the VoIP-specific Honeypot (see section 4.2).

4. VOIP-SPECIFIC HONEYPOT

In literature, a Honeypot is a trap set to detect, deflect and monitor attacks to information systems. Generally it consists of a computer, data or a network site that appears to be part of a network but which is actually isolated and monitored. A Honeypot has a specific value in attack detection and deflection. Honeypots usually are specific systems that normally should not see any legitimate traffic or activity. Whatever activity is seen on a Honeypot it can be interpreted as malicious or unauthorized.

The Honeypot concept can become very useful as specific component of a VoIP intrusion and prevention architecture. We brought the Honeypot concept into the VoIP world de-
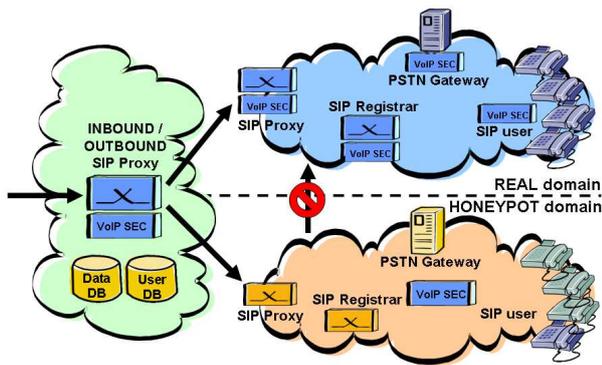
Figure 1: Network Architecture

veloping a complete parallel VoIP infrastructure completely logically and physically separated by the real one where we continuously monitor activity. Physical separation is necessary in order to avoid that an attacker breaking into the VoIP-specific Honeypot is able to attack the real VoIP infrastructure from there.

The key features of the VoIP-specific Honeypot are the mitigation of the SPIT threat with low cost infrastructure as described in 4.1 and the fact that such an infrastructure well complements all the other possible intrusion detection and prevention by information gathering useful to improve error rate of other methodologies as described in 4.2.

4.1 Architecture and Implementation

The VoIP-specific Honeypot depicted in Figure 2 is composed of standard open-source VoIP components based on the SIP protocol. The main goal of a Honeypot is to attract attacks into a secured and observed environment for analyzing their attacking schemes and deduce new kind of prevention methods against them. To reach this goal the Honeypot has to offer attractive services that are worth to be attacked. These services were simulated because of performance and resources issues. A SIP Honeypot has to simulate a whole SIP network, which can offer many fetching values:
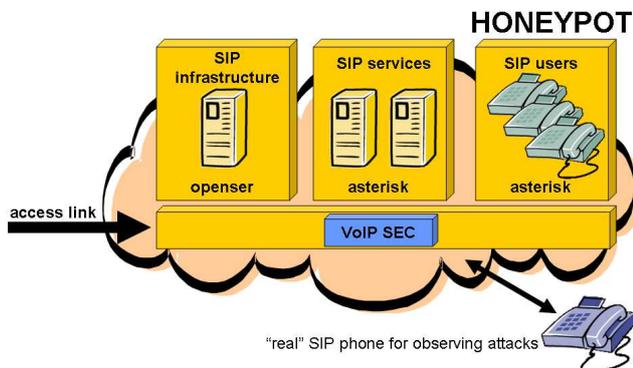
- SIP components
- SIP services
- SIP users

Figure 2: Honeypot Architecture

For the implementation of the VoIP-specific Honeypot we used a mix of well known software:

- Openser[2]: an open source SIP proxy server software. The openser advanced routing logic allows to configure complex routing schemes and its modular design can be extended with additional modules for more advanced services;

- Asterisk[3]: an open source PBX. Asterisk offers a wide range of services and applications (e.g. gateway functionalities, mailboxes or Interactive Voice Response, IVR, applications) allowing a wide range of PBX scenarios emulation on a Honeypot.

The VoIP-specific Honeypot emulates a real VoIP network as follows:

- the SIP infrastructure is emulated by Openser. An instance of Openser is configured in the VoIP-specific Honeypot. Multiple instances can be used to build more complex SIP networks. Openser is configured with a random call-dispatcher for handling unknown callees. Callers initiating sessions to unknown callees gets randomly connected to the emulated services and users.

- the SIP services are emulated by Asterisk. With dedicated numbers (Asterisk extensions) the attacker gets redirected to voice-mail boxes (e.g. the SIP-PSTN gateway service is a simple prerecorded messages that signals the temporal unavailability of such a service.

- the SIP users are emulated by Asterisk mailboxes as well. Optional SIP phones can be implemented to mirror every incoming call for observation purposes.

4.2 Detection and Prevention Scheme

In order to stimulate attacks towards the Honeypot and to trap as much attackers as possible we prepared scripts that are going to publish systematically fake user URIs (in web pages, emails, newsgroups, etc.) registered to the Honeypot SIP proxy server in a way that is either not possible to be seen by humans (e.g. URIs written in white over a white background in a web-page) or easily recognizable by humans as URIs not corresponding to real users (e.g. URIs with the word do-not-call) but at the same time that can be easily harvested by bot-nets automatically looking for semantics of URIs. The objective is, with such a big number of user URIs advertised on the Internet, to make the VoIP-specific Honeypot and its users target of social attacks like SPIT and/or Vishing (since bot-nets that harvested the URIs are not going to make distinctions on them).

The rationale behind such VoIP deflectors (VoIP-specific Honeypot users) is that they should normally not experience traffic activity and therefore every activity seen by these deflectors can be interpreted as a malicious one. If any of these deflectors receives a call this is considered as an indication that the sender is an initiator of SPIT/Vishing attacks.

Information gathered by such a monitoring activity is then used as input to the prevention system reacting to attacks. A simple reaction is to add users that initiated activities to the

---

VoIP-specific Honeypot to a black list that can be temporary or permanent. The users in such a list are not allowed to initiate communications to the real infrastructure. For this purpose we programmed the VoIP-specific Honeypot to publish its results in a database accessible for reading to the real infrastructure. VoIP-specific Honeypot federations sharing such monitoring activities among multiple site for the purpose of improved detection is currently subject of an on-going work. The idea in this case would be to have federations of domains that share the VoIP-specific Honeypot activity reports in order to build a list of untrusted users (users seen on multiple VoIP-specific Honeypots are more likely to be malicious and not to have made an error in dialing).

The big number of fake user URIs has also a second objective which is the deflection of attacks, as a matter of fact if the attacker harvests for SIP URIs (either from the web or doing a user enumeration attack) it will retrieve not only the real x ones but also the fake y ones. In this case the mitigation percentage will be proportional to the ratio between real and fake users y/(x+y) since the SPIT/Vishing calls will be distributed among all users (real and fake ones).

The VoIP-specific Honeypot has not only the objective of monitoring intrusions but has also to gather additional features of attacks characteristics for intrusion detection and prevention in the case of SPIT/Vishing attacks. The Honeypot software (mainly the terminal part) is composed of different answering machine that can:

- register the call and analyze it later: messages of the same length can be classified as SPIT/Vishing attacks, messages can be analyzed and compared to messages in the voice mailbox of real users. Such information can be used to process messages in the real voice mailbox and sort accordingly voice mailbox of users dividing messages in good ones and SPIT/Vishing ones in order to facilitate users in listening to their voice messages.

- deeper interaction with the originator in order to gather additional information on the source in order to characterize it and input such results to other identification mechanisms. An example of this are the fingerprint checks (IP addresses, software client used to initiate the call, IP path towards the source, other layer 4 ports the initiator may have open that can indicate it is initiating many calls in parallel, etc.) detailed in [9] or an extension of the audio turing test detailed in [15] aimed to fingerprint the audio sent by machines. The output of such fingerprint can then be used by other detection mechanisms to further reduce the error rate (calls with characteristics observed already by the deflectors will be detected with higher probability).

- make the blocking (suppression in legal terms) of communications compliant to the suppression of communication law. The idea is to send the calls that are considered malicious by the VoIP Security Event Correlator detailed in 5 to the deflector infrastructure instead of blocking them. In this case the communication will not be suppressed and further recorded as if it were in a separate voice mailbox being available for the callee; at the same time these calls will be used to further refine the detection mechanism.

5. VOIP SECURITY EVENT CORRELATION (SEC)

As mentioned before, VoIP systems are distributed by nature, thus it is obvious that studying traces of just one VoIP user agent or server does not give an overall picture of what is happening to the whole system. An attack involving several domains could result in not being detected just monitoring the single traces at each domains. Aggregating traces from different sources is vital to detect useful signatures. Our approach is to deploy at every point of interest (proxy, user agent, gateways) a first layer of monitoring composed of threading and correlation. Then, events of interest are given as input from different sources to a central correlator. The event correlation hierarchy is depicted in figure 3, in order to better explain the advantage of such an approach the following example is given:

Malicious gateway.
In a normal scenario, a user sets up a call towards the PSTN by contacting a call agent. The call agent controls the gateway by using the MGCP protocol. It opens a media trunk in the gateway using a CRCX command letting the user send the RTP flow according to the specification included in the SDP protocol body. In a toll fraud attack scenario, the malicious user aims to bypass billing. The malicious user could set up the call normally, then immediately sends a BYE to the call agent. The call agent releases the call by a DLCX command, the accounting procedure stops upon receiving a 200 OK from the gateway. But if the gateway was instructed by the malicious user to not respond to the call agent and to send falsified notifications, the malicious user will continue to use the media trunk without being billed for the usage. Such intrusions could not be detected if only the MGCP call agent trace or the end point one are monitored but by a central Security Event Correlator (SEC) can detect RTP packets received by the endpoint after a DLCX command is sent by the call agent.

5.1 Event generation

Most VoIP servers and phones provide protocol logs either to standard output or as text files in order to help the debugging process. Asterisk, OpenSER and Kphone[4] are some examples. However, the purpose of an intrusion detection system is not debugging VoIP agents but detecting security incidents.

The task of the event generator located at each call agent (MGCP Call Agent, SIP Proxy, etc.) is to extract predefined fields from the protocol message and to add a timestamp. SIP and MGCP signaling protocols have completely ASCII oriented headers and bodies (SDP). In case of binary based protocols as RTP for media transport and H323 for signaling, the event generator has to properly parse the relevant fields to ASCII to be coherent. Such an event generator can be configured to allow multi-layer analysis by including IP and TCP/UDP information and has all reassembly capabilities to deal with IP fragmentation, TCP segmentation and protocols decoding in order to recognize protocol messages. To better explain the event generation, the following array is given as example when a SIP INVITE message is considered:
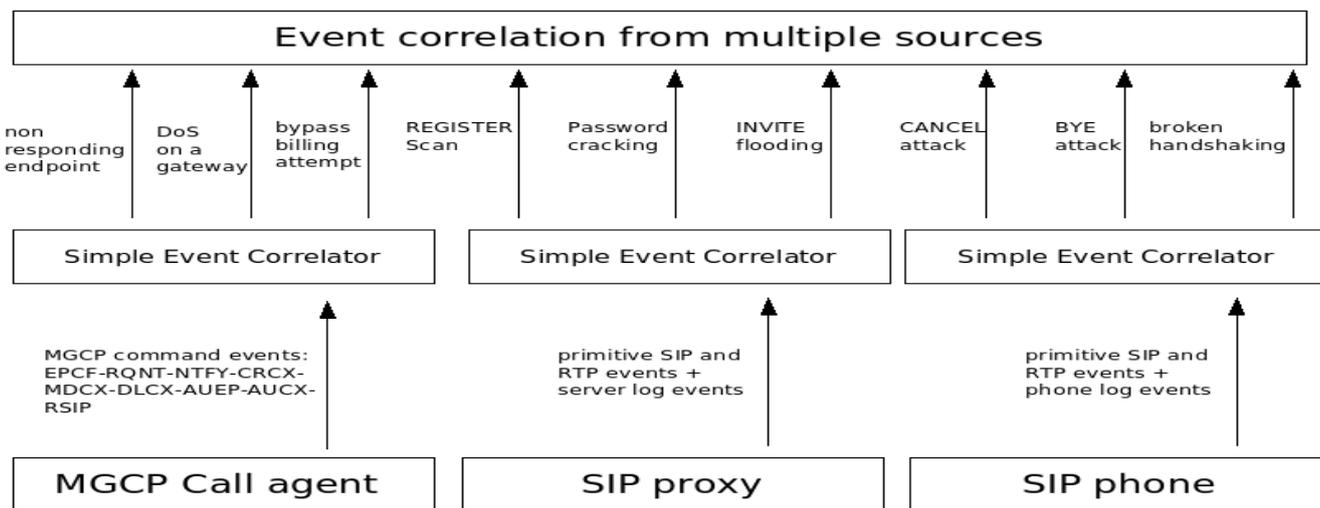
---

[4] http://sourceforge.net/projects/kphone

Figure 3: Two layers of event correlation

Arrival time	Nov 7 2006 09:06:29
IP source	192.168.1.108
IP destination	192.168.1.4
Source port	UDP/5060
Destination port	UDP/5060
SIP header	
Via	192.168.1.108
Via	192.168.1.106
contact	192.168.1.106
From	sip:5005@192.168.1.6
To	sip:2002@192.168.1.4
SDP body	
Owner	192.168.1.106
Media	audio
protocol	RTP/AVP
port	49152

The important fields associated with a RTP packet are instead shown below:

Arrival time	Nov 7 2006 09:06:53
IP source	192.168.1.106
IP destination	192.168.1.4
Source port	49154
Destination port	17138
RTP header	
Seq. Number	23086
Time stamp	0
SSRC	273598425

In order to detect intrusions in VoIP architectures a model-building of normal behavior or malicious activities is of paramount importance. Appropriate specification of an event content and its relationship to other events in the time are the key requirements to build complex events from primitive ones. We show the importance of these requirements by mentioning some examples:

Event threading.

Besides primitive events, building patterns of activities is mandatory for security intrusion monitoring. User profiling is a possible example where user normal patterns need to be defined (e.g. user 'A' makes 5 calls on average during weekdays and 2 calls on average during weekends). To model such patterns, it is easier to deal with primitive call events than with basic protocol messages. A PBX has already support for such events in its Call Detail Records (CDRs) but a SIP proxy log needs a dedicated application matching every INVITE with the corresponding BYE to identify a session as follows:

Call time	Mon oct 23 2006 14:08:07
Caller	A@Inria.fr
Contact	100.101.102.103
Callee	Bob@loria.fr
Duration	00:02:19
Bill-duration	00:01:88
Disposition	Answered

Also, when searching for some attack patterns, the signature of an attack could not be detected by matching one single event but by correlating a sequence of events appearing to be normal if analyzed separately. For example, in a flooding DoS we do like to bound a large number of similar events as just one alarming event and suppress not necessary redundancy by just adding the number of messages from which the alarming event was composed.

Temporal restrictions.

Scheduling restrictions and the inter-arrival timing of events are included in the temporal restrictions considered by VoIP SEC.

Scheduling restrictions are very useful in monitoring tasks. A MGCP call agent is supposed to send an AUdit Endpoint command (AUEP) to each endpoint under its supervision. Assuming that the normal activity is to make audits at fixed dates (e.g. every hour o'clock), any fluctuation in the call agent behavior can be detected by noticing the absence of such events at specified times.

The inter-arrival time of events is even more relevant. A short inter-arrival time of requests to a SIP target characterizes a flooding DoS, a high rate of appearance of '404 Not Found' responses characterizes a domain enumeration

and a high rate of appearance of '403 Forbidden' responses characterizes a password cracking attack.

To clarify our approach regarding temporal restrictions let us take the following attack scenario:

1. One SIP caller from Internet attempts to call a PSTN number through a gateway controlled by the MGCP protocol;

2. the MGCP call agent receives the INVITE, translates it into SS7 signaling to ring the PSTN phone;

3. once the call is answered, the MGCP call agent responds to the SIP caller with a 200 OK, it sends a CRCX (create connection) command to one end point in the media gateway to link the RTP flow and a voice trunk;

4. the caller has to acknowledge such a command with the final response (200 OK). A malicious caller may do not complete the handshake to achieve one of two purposes:

   - bypass billing: if the SIP call agent is compromised and it waits for the ACK to consider that the session is open and to start billing, the caller can use the media connection without being charged for it.

   - resource exhaustion DoS: if the caller uses an army of bots initiating a set of messages aimed to open media connections bypassing billing without sending RTP data, the call agent will be overloaded with a high number of paralyzed state machines.

A solution addressing this attack scenario is to impose a temporal restriction on the ACK message that must be received within a few round trip times after the 200 OK.

5.2 Event Correlation using SEC

After investigating available software for event correlation, the most suitable was found to be the open source and platform independent SEC [5]. SEC fulfills VoIP events modeling requirements using static rules, accepting input from text streams named pipes and generating output events, log messages or executing shell commands. Each rule in SEC contains an event matching condition based on a Perl regular expression, an actions list, and optionally a boolean context that permits or not the application of the rule.

SEC is a lightweight online monitoring tool initially invented to fill the gap between homegrown and commercial event correlation solutions, and it already proved its efficiency in several domains as network management, intrusion detection systems, system monitoring and fraud detection. However, to the authors best knowledge, with new generation Internet applications having specific properties as VoIP, it is not been tested yet. Some VoIP threats have known patterns that can be detected by signatures matching. However, the misuse detection solution based on signatures can not accommodate previously unknown attacks and is not effective to detect service abuse. Anomaly-based intrusion detection is efficient to detect fraudulent usage, virus propagation and social threats like SPIT. In the anomaly-based approach, a statistical profile is created for normal activity of subjects

(e.g. user, call session, SIP server) with respect to objects (e.g. network resources, gateway trunks, server CPU and memory). This statistical profile generates anomaly records when deviations from normal behavior are occurring. SEC rules are efficient to detect attack signatures, and statistical profiles of users, group of users and traffic can be defined to monitor behavior anomalies. In table 5.2, examples of both approaches are given to show how different types of SEC rules can be designed.
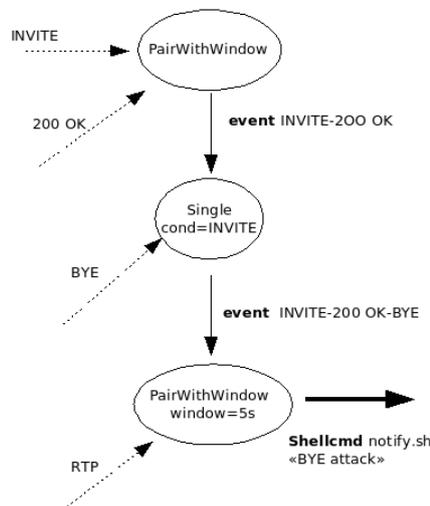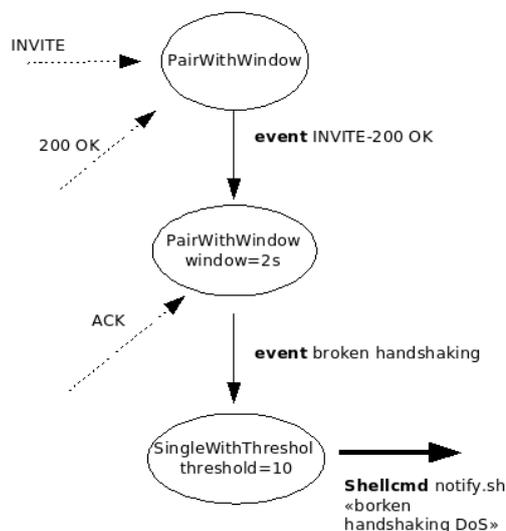
Figure 4: Diagram of SEC ruleset to detect BYE attack

Figure 5: Diagram of SEC ruleset to detect broken handshaking flooding

5.3 Implementation

We have implemented a module in the OpenSER code that copies the SIP messages routed by the server towards a parser that is written with SEC rules. A few lines in the OpenSER configuration file have been added after installation for this scope. The SEC parser creates a line with fields

[5] http://kodu.neti.ee/~risto/sec/

Table 1: Using SEC to build efficient correlation rules

| Attack description	Detection scheme
Signature-based intrusion detection	
DoS using BYE attack: In a call between A and B, a malicious entity C sends a DoS on A by prematurely tearing down the call with a crafted BYE message to A. If A receives RTP packets from B after receiving the BYE, this means that B was unaware of the BYE sent. Similar attacks are call hijacking, instant messenger hijacking and RTP play out attack.	Three rules of type PairWithWindow are needed to detect such an attack. PairWithWindow is a composed SEC rule that after a first defined event arrives, waits for t seconds for other defined events, and executes one list of actions if the second event arrived in time, or another list of actions if the window timeout expires. The detection scheme for this attack consists on: • one PairWithWindow rule to match an INVITE with the corresponding 200 OK and generate an INVITE-2OO OK event with the Call-ID, From and To tags, the media IP and port of the caller (in the INVITE) and the media IP and port of the callee (in the 200 OK); • the second rule to match the INVITE-200 OK event with the corresponding BYE event and generate an INVITE-2OO OK-BYE event; • the third PairWithWindow rule to match the INVITE-2OO OK-BYE event with a RTP event sent from the caller to the callee that occurs in the time window and in this case write an alarms to the GUI as shown in figure 4.
DoS using broken handshaking: this attack is based on broken SIP handshaking where the attacker sends an INVITE request and then ignores the 200 OK response refusing to send the ACK. The attacker (or an army of bots) proceeds with a large number of broken initiations in order to exhaust the target performances.	one rule of type PairWithWindow and one rule of type SingleWithThreshold are needed to detect this attack. SingleWithThreshold is a composed SEC rule that counts matching events in a window of t seconds, and if a defined threshold is exceeded executes a list of actions, otherwise it waits the expiration of the time window to execute another list. The detection scheme for this attack consists on: • one PairWithWindow rule to match the INVITE with the corresponding 200 OK and generate an INVITE-2OO OK event with the Call-ID, From and To tags; • the second rule to match the INVITE-2OO OK event with the corresponding ACK event. If no ACK is received within the time window, the rule generates a `broken handshaking` event; • the SingleWithThreshold rule counts the `broken handshaking` event in a time window and writes an alarm to the GUI in case of a defined number is exceeded as shown in figure 5.
Anomaly-based intrusion detection	
User profile: One method of building an account profile is to make use of histograms. A day is divided into bins of specified time (e.g. one hour). For each bin a predefined metric is calculated (e.g. number of calls, number of different recipients, average duration of a call) matching predefined events (e.g. call). In the learning phase (e.g. a month), daily statistics are built to extract a long term account profile (e.g. daily average of the number of calls for each bin). In the detecting stage (e.g. a day), a short term profile is compared to the long term one by using an appropriate distance function (e.g. Euclidean distance, quadratic distance, Mahalanobis distance). A recent profile which is quite different from the long term one indicates possible misuse. On the other hand, long term profiles can be compared to different accounts to group them into classes. This is of high importance since a class of VoIP bots send SPIT calls can be detected if a known bot profile falls into the same class. Another method is to study non stationary features of an account, for example the distribution of calls over all callees or the shape of the calleesŠ list size over all dialed calls. By comparing changes of a distribution over the time by using of an appropriate distance function (e.g. Hellinger distance), sudden bursts may be detected and treated as abnormalities.	In addition to appropriate scripts to build up such a profile for a given user, SEC rules can be of importance to manage the process. Calendar is a basic SEC rule that gets activated at specific times and executes a list of actions using a UNIX cron-like syntax. SingleWithScript is a composed SEC rule that matches input event and executes a list of actions depending form the return value of an external script. For a complete view of SEC design and allowed actions, please refer to [1]. The detection of abnormal behaviors can be implemented using SEC as follow: • a rule of type Calendar gets activated at specified times to launch a `time to compare` event. • a rule of type SingleWithScript catches the event and launches an external script that has access to short-term and long-term profiles. If the external script returns a zero value for a weak similarity, the corresponding action will notify about an abnormal behavior. The corresponding rule set is depicted in figure 6.

Table 2: Size of rulesets detecting various attacks

Name of attack	Number of rules	Name of attack	Number of rules
Request flooding	2	BYE attack	3
Broken handshaking flooding	3	CANCEL attack	3
REGISTER scan	4	Malicious gateway	4
password cracking	2	SPIT	4

```
type=Calendar
time=0 0 * * *
desc=time to compare
action= event 0 %s user1

type= SingleWithScript
ptype=RegExp
pattern=time to compare (\S+)
script= /usr/bin/perl compare.pl $1
desc=compare long-term and short-term profiles  for user $1
action= shellcmd notify.sh ''abnormal behavior user $1''
```

Figure 6: SEC ruleset to manage profile comparison
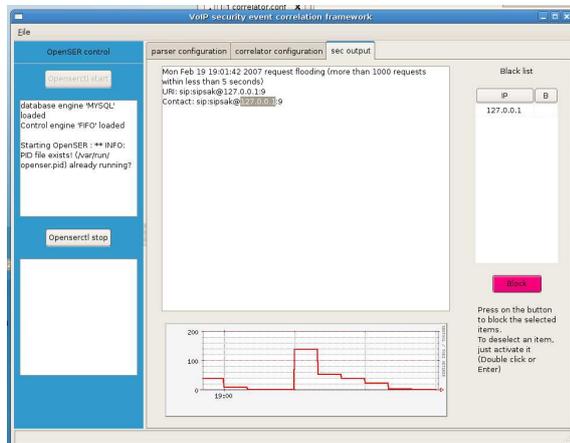
Figure 7: Detection of DoS attack

of interest (From, Contact, the request type) for each SIP message. These lines are treated by a correlator which is written with SEC rules too. The correlator writes alarms of detected misuses to a GUI as in Figure 7. The graph shown at the bottom of the figure is generated with a specific Round Robin database library. The alarm corresponds to a DoS peak. In the case the attack is detected and persists over time, human decision can select the IP address of the detected attacker and add it to the black list in order to block it either temporarily or definitely.

Table 2 shows the attacks implemented so far and the number of SEC rules written for their detection. The scalability of the solution is clear when looking at the small number of rules sufficient to detect different series of attacks. The performance of the VoIP Security Event Correlator built in this way is therefore promising but was not quantitatively tested. In addition to OpenSER extensions, the current work is focused on implementing a "local" event correlator in Asterisk and in additional softphones (Kphone) and a central correlation engine as a second layer of detec-

tion.

6. RELATED WORKS

Intrusion detection systems are a second line of defense behind intrusion prevention mechanisms as password authentication and firewalls. One of the earlier works in this domain is the model proposed in [2]. The authors of [4] shows the necessity of domain knowledge in specific IDSs especially with web-based applications. Security is of great interest in the new VoIP generation design so several intrusion detection approaches are proposed in response to different threats. Scidive [14] uses signature-based statefull and cross protocol schemes. We proposed a statistical frame work based on Bayes model to recognize normal and attack SIP traffic classes in [6]. SEC was proposed in [12] as a lightweight event correlator that can serve different applications ranging from log file and system monitoring to fraud detection, network management and intrusion detection and a good overview on it can be found in [10]. Recent papers on the detection of social attacks proposed blacklist/graylist type solutions [11] and centralized network entities [5]. Previous work of a subset of the authors [8] considered network level plugins for Snort capable of detecting SPIT attacks. An interesting idea of SPIT detection based on device fingerprinting is exposed in [15]. A first description of a VoIP honeypot can be found in [7] where a subset of the current authors proposed network level SIP honeypot.

7. CONCLUSION

We have presented in this paper a holistic approach for VoIP security monitoring. The key components of our solution are a VoIP honeypot and a SIP level event correlation engine. This solution is capable to defend against both brute force denial of service attacks as well as more stealthy social type (SPIT, Vishing). We leveraged the capabilities of SEC to constitute the technical tool in building efficient VoIP IDS and showed the feasibility by developing a prototype. We extended in this paper the notion of a honeypot towards VoIP applications and showed how social attacks can be mitigated. We have implemented and tested our solution in a testbed environment, but more real life tests and performance evaluation will be done in the future on a VoIP network. Future work will address also the extension of the current event correlation techniques towards machine learning inspired paradigms.

8. REFERENCES

[1] J. Brown. *Working with SEC - the Simple Event Correlator*. http://sixshooter.v6.thrupoint.net/ SEC-examples/article.html;http://sixshooter.v6. thrupoint.net/SEC-examples/article-part2.html.

[2] D. E. Denning. An Intrusion-Detection Model. In *IEEE Symposium on Security and Privacy*, pages 118–133. IEEE Computer Society Press, Apr 1986.

[3] D. Endler and M. Collier. *Hacking VoIP Exposed.* McGraw-Hill Osborne Media, 2006.

[4] C. Krügel, T. Toth, and E. Kirda. Service specific anomaly detection for network intrusion detection. In *SAC '02: Proceedings of the 2002 ACM symposium on Applied computing*, pages 201–208, New York, NY, USA, 2002. ACM Press.

[5] B. Mathieu, Y. Gourhant, and Q. Loudier. SPIT mitigation by a network level anti SPIT entity. In *Third annual security workshop (VSW'06).* ACM Press, June 2006.

[6] M. Nassar, R. State, and O. Festor. Intrusion detections mechanisms for VoIP applications. In *Third annual security workshop (VSW'06).* ACM Press, June 2006.

[7] M. Nassar, R. State, and O. Festor. VoIP Honeypot Architecture. In *Proc. of 10 th. IEEE/IFIP Symposium on Integrated Management*, June 2007.

[8] S. Niccolini. SPIT and SPIM. In *Third annual security workshop (VSW'06).* ACM Press, June 2006.

[9] J. Quittek. Detecting SPIT calls by checking communication patterns. In *IEEE ICC 2007*, Jun 2007.

[10] J. P. Rouillard. Real-tile Logfile Analysis Using the Simple Event Correlator (SEC). In *18th USENIX System Administration Conference (LISA '04) Proccedings*, pages 133–149, November 2004.

[11] D. Shin and C. Shim. Voice SPAM Control with Gray Leveling. In *2nd Workshop on Securing Voice over IP*, June 2005.

[12] R. Vaarandi. SEC - A Lightweight Event Correltion Tool. In *Proceedings of the 2002 IEEE Workshop on IP operations and Management*, number 0-7803-7658-7, pages 111–115, October 2002.

[13] VoIPSA. VoIP security and privacy threat taxonomy. Public Realease 1.0, Oct 2005. `http://www.voipsa.org/Activities/VOIPSA_Threat_Taxonomy_0.1.pdf`.

[14] Y. Wu, S. Bagchi, S. Garg, N. Singh, and T. K. Tsai. SCIDIVE: A Stateful and Cross Protocol Intrusion Detection Architecture for Voice-over-IP Environments. In *International Conference on Dependable Systems and Networks (DSN 2004)*, pages 433–442. IEEE Computer Society, Jun 2004.

[15] H. Yan, K. Sripanidkulchai, H. Zhang, Z. Shae, and D. Saha. Incorporating Active Fingerprinting into SPIT Prevention Systems. In *Third annual security workshop (VSW'06).* ACM Press, June 2006.