An Adaptive Policy-Based Approach to SPIT Management

Yannis Soupionis, Stelios Dritsas, and Dimitris Gritzalis

Information Security and Critical Infrastructure Protection Research Group,
Dept. of Informatics, Athens University of Economics and Business,
76 Patission Ave., Athens, GR-10434, Greece
{jsoup,sdritsas,dgrit}@aueb.gr
http://www.cis.aueb.gr

Abstract. Voice over IP (VoIP) is a key enabling technology, which provides new ways of communication. VoIP technologies take advantage of existing data networks to provide inexpensive voice communications worldwide as a promising alternative to the traditional telephone service. At the same time, VoIP provides the means for transmitting bulk unsolicited calls, namely SPam over Internet Telephony (SPIT). SPIT is, up to a given extend, similar to email spam. However, it is expected to be more frustrating because of the real-time processing requirements of voice calls. In this paper we set the foundations of an adaptive approach that handles SPIT through a policy-based management approach (aSPM). aSPM incorporates a set of rules for SPIT attacks detections, together with appropriate actions and controls that should be enforced so as to counter these attacks. Furthermore, the policy is formally described through an XML schema, which refers to both, the attack detection rules, and the corresponding defense actions.

Keywords: VoIP, SPIT, Attack Graphs, Attack Trees, Policy, Rules, Actions.

1 Introduction and Related Work

The explosive growth of the Internet has introduced a wide array of new technological advances and more sophisticated end-user services. Development in data networks facilitated the introduction of VoIP technologies, which have been increasingly penetrating the telephony market in the last years. VoIP advantages include seamless integration with the existing data networks, portability, accessibility, and convergence of telephone networks. These are some of the reasons that make VoIP an attractive and advantageous network technology.

Currently, VoIP implementations are mainly based on the Session Initiation Protocol (SIP) [1], which tends to be the dominant protocol in VoIP environments. However, the use of Internet Telephony in accordance with the vulnerabilities posed by its underlying infrastructure, i.e. the Internet and the SIP

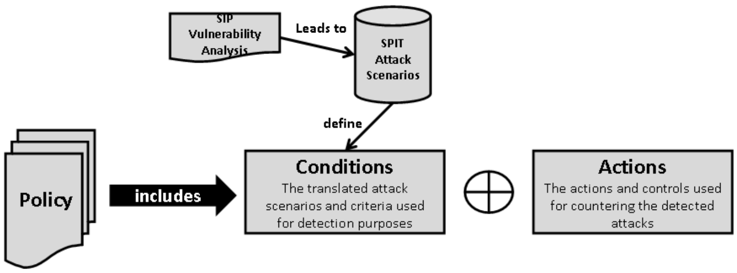
SIP Vulnerability Analysis
Leads to
SPIT Attack Scenarios
define
Policy
includes
Conditions
The translated attack scenarios and criteria used for detection purposes
Actions
The actions and controls used for countering the detected attacks

Fig. 1. A macroscopic view of the aSPM approach

protocol, also facilitates the exploitation of new threats and vulnerabilities. For instance, low- or zero-cost calls and zero-cost instance messages, combined with the pervasiveness of Internet, could be an attractive tool for malicious users, e.g., spammers, to make bulk unsolicited calls and/or send bulk unsolicited instant messages. This situation introduced a new form of spam, which - in the case of VoIP environments - is called Spam over Internet Telephony (SPIT) [2,3]. If there are no means to counter SPIT effectively,then an unfortunate situation will probably arise, where the use of Internet Telephony would be a synonym to frustration, and not to convenience and cost-effectiveness.

In this paper we propose a policy-based approach, as a means to manage the SPIT phenomenon in a holistic way. The approach is primarily based on a SIP protocol threat and vulnerability analysis. A result of this analysis was the identification of a series of attack scenarios. In turn, the attack scenarios were facilitated in an effort to define SPIT detection rules. These rules led to the identification and description of specific actions and controls for handling, countering, and mitigating SPIT attacks. Finally, an XML schema was used as a means for both, first, describing the detection rules, and, second, for specifying the SPIT handling controls and actions. Figure 1 depicts the functionality of this approach.

The paper is organized as follows: First, we illustrate related work on SPIT fighting and management. Then, we present the main parameters of the SPIT phenomenon, together with the notion of electronic policies. In section 4 we describe, in a generic way, the proposed policy-based SPIT management system (aSPM). In section 5, we briefly present SPIT-related vulnerabilities. In the following section, we describe how we realize SPIT attack scenarios, based on attack graphs. In section 7, we analyze aSPM further, based on an appropriate XML schema, together with the conditions that might indicate a SPIT attack, and a set of counter actions. In section 8, we propose how aSPM can be practically applied to a SIP environment. Finally, we arrived to a number of conclusions, and our plans for future work.

Current methodologies for developing anti-SPIT policies are described in more or less abstract level. As a result, they focus mainly on high level aspects of security, i.e. user preferences, while they leave aside technical aspects, such as authentication and authorization requirements, etc.

Two of the more often referred to papers on anti-SPIT policy are proposed as Internet drafts (IETF). The first one [4] proposes an authorization policy and recommends an XML structure, which identifies possible SPIT and suggests countermeasures. The main drawback of it is that the identification is based mainly on users URI and not on other SIP protocol aspects. Furthermore, it does not include the recommended rules and conditions. The second one [5] is a Call Processing Language (CPL), which describes and controls Internet telephony services. It is developed for either network servers, or user agents. It provides an XML schema, as well as the proposed values of some of its element. The weakness of this approach is that it is quite generic. It is focused on how one can represent VoIP services, and ignores the SPIT phenomenon and how it can be prevented.

Our approach aims at reducing the SPIT threat. The means for doing so is a policy-based management system, relied on well-defined criteria and countermeasures, which are applied directly to the SIP messages. Moreover, this approach is independent of the application that is used in the VoIP Infrastructure.

2 SPIT Phenomenon

2.1 VoIP and SPIT

VoIP implementations are usually based on the Session Initiation Protocol (SIP) [1]. SIP is an application layer protocol that is used to create, maintain, and terminate multimedia sessions. The basic SIP entities, which support its functionality, are (a) User Agents (UA), which act as communication end-points, and (b) SIP servers, which help and support the SIP sessions.

SPIT is a new type of threat in VoIP infrastructures. It is defined as a set of bulk unsolicited voice calls or instant messages. SPIT has three different types, namely [3]: (a) call SPIT, (b) instant message SPIT, and (c) presence SPIT. SPIT is expected to become, sooner or later, attractive to malicious users, thus making the further growth of VoIP technology practically challenging. Managing SPIT requires: (a) appropriate criteria for SPIT detection, as well as (b) actions, controls, and countermeasures for SPIT handling. Such a management capability, in terms of detection and handling, is hard to attain, mainly due to the real-time nature of VoIP communications. The problem becomes worse, as the techniques which are currently adopted for anti-spam purposes (i.e. content analysis based on Bayesian filters, or approaches [6] which aim at preventing SPIT by recognizing voice communication patterns, etc.) are not fully applicable to the SPIT context.

2.2 Policies

A VoIP infrastructure is actually a software-based application system that aims to assist users to communicate with each other. However, due to its inherent characteristics, it may also help malicious users to exploit it and make low- or zero-cost unsolicited calls.

In this paper we propose an adaptive policy-based SPIT management system, which can: (a) identify SPIT calls/messages and (b) provide appropriate actions (countermeasures) during the initiation handshake of a VoIP call. The adaptation property is achieved by providing not only the option of adding new conditions and/or actions, but also the ability to administrators of each VoIP domain to choose the appropriate rules, according to their preferences and needs.

Policies can be sorted into two basic types, namely [7]: authorization policies and obligation policies. Authorization policies are used to define access rights for a subject (management agent, user, or role). They can be either positive (permit action on target object), or negative (forbid action on target object). As such, authorization policies are used to define access control rules implemented by several types of mechanisms in a network security system, such as packet filters. Obligation policies are event-triggered condition-action rules are used to define what kind of activities a subject (human or automated manager components) must perform on objects in the target domain. In the network security context, obligation policies can be used to specify the functionality of mechanisms, such as intrusion detection systems (IDS).

We consider the anti-SPIT policy as an obligation policy. It facilitates an existing set of relevant rules and enables SPIT handling, i.e., refers to the actions that should be considered whenever a SPIT call/message is detected. Policy rules define which behavior is desired (legal) in a VoIP system. They do not describe the actions and event sequences that actually produce desired or undesired behavior. Therefore, policy rules alone are not sufficient to model every (behavioral) aspect of a VoIP system. Therefore, a policy rule set can only be assessed and sensibly interpreted in combination with adequate knowledge, regarding its embedding context. For this reason, a series of attack scenarios could help for modeling better the VoIP context.

3 Methodology

The proposed SPIT management methodology is depicted, in a functional manner, in Figure 2.

The first step of the methodology aims at an in-depth examination and analysis of the SIP protocol, in terms of SPIT. The scope of the analysis is to identify the SIP-related SPIT vulnerabilities, as well as the methods used by the attackers (spitters). The result of the SIP analysis was a number of well-defined SPIT-related threats and vulnerabilities, in accordance with the SIP RFC [1]. In the second step, the identified vulnerabilities are divided in categories. Such a categorization can help the VoIP system administrator recognize and enforce specific policy rules to specific entities, according to the communication segment each one belongs to. The third step aims at developing the attack scenarios. These scenarios were essential, so as to produce the appropriate rules for the policy. The development of scenarios is a two-step procedure: (a) a SPIT-oriented attack graph was designed, based on the identified vulnerabilities (its underlying attack trees were also built), and (b) a set of SPIT attack scenarios was produced, having the corresponding attack graph as input. The next step aims at

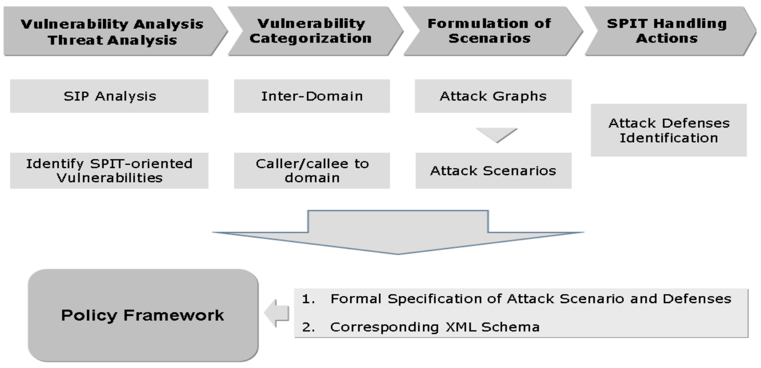
Vulnerability Analysis Threat Analysis
Vulnerability Categorization
Formulation of Scenarios
SPIT Handling Actions
SIP Analysis
Identify SPIT-oriented Vulnerabilities
Inter-Domain
Caller/callee to domain
Attack Graphs
Attack Scenarios
Attack Defenses Identification
Policy Framework
1. Formal Specification of Attack Scenario and Defenses
2. Corresponding XML Schema

Fig. 2. aSPM: Development methodology

describing a set of SPIT handling actions and controls, so as to counter each and every attack scenario. The selection of the essential actions was, mainly, based on SIP messages [8,9]. The actions are SIP messages, because the policy should: (a) be transparent to the administrators and users, and (b) keep to a minimum the participation of other applications during message handling. The final step aims at formally combining the attack scenarios with the proposed countermeasures (actions). In order to do so, the XML language was selected and used. The development of an XML schema not only facilitated the definition of a formal anti-SPIT policy, but also generated a flexible policy description, which is adoptable by most SIP infrastructures.

4 Threat Analysis - Vulnerability Analysis and Categorization

Although several similarities exist between spam and SPIT, the real-time nature of VoIP sessions and services, in contrast with the store-and-forward functionality of email technology, force to consider new ways for SPIT handling.

It is evident that it is more efficient to deal with SPIT in the signaling phase, i.e., where the SIP protocol is applied, than to process the content of an instant call. Thus, an in-depth analysis of the SIP protocol was carried out, in order to identify the SPIT-related SIP vulnerabilities and threats [10,11]. The set of vulnerabilities proposed in these papers were: (a) related to SIP protocol vulnerabilities, (b) due to SIP RFC optional recommendations, (c) related to interoperability with other protocols and (d) due to more generic security issues. For the analysis we considered the threats, which were derived from the first two categories. Table 1 depicts the identified vulnerabilities, which can be exploited by a spitter. The categorization of vulnerabilities was performed by taking into account that there are multiple points of communication, where the anti-SPIT policy actions could be enforced. The points of communication refer to all the entities that participate to a SIP session establishment, i.e., the intermediate

Table 1. SPIT-related SIP vulnerabilities

List of vulnerabilities	
General vulnerabilities	
Listening to a multicast address Sending messages to multicast addresses	Population of "active" addresses. Exploitation of forking proxies.
Exploitation of messages and header fields structure	
Request Messages: INVITE and ACK Response Messages	Request Messages: MESSAGE
Header fields of messages	
Subject Contact and To Warning Content-Type Monitoring traffic near SIP servers Contacting a redirect server with ambiguous requests Misuse of stateless servers	From Retry After Content-Disposition Priority Sending Ambiguous Requests to Proxies Throwaway SIP accounts Anonymous SIP servers and back-to-back user agents
Exploitation of messages headers fields	
Alert-Info Error-Info Port scanning of well-known SIP ports Exploitation of the record-route header field	Call-Info Exploitation of registrars Exploitation of re-INVITES messages

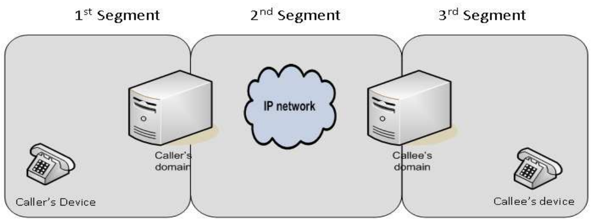

Fig. 3. Communication segments

domain proxies, as well as the proxies of the caller and the callee. The proposed segments, presented in Figure 3, are: (1) the communication part which lies between the callers device and the domain proxy that serves the caller, (2) the communication part that lies between proxies or redirect servers, and (3) the communications part that lies between the callees device and the domain proxy of the callee.

The first segment is the point where the SIP session establishment requests start. This segment also receives responses from the intermediate or final servers that serve the caller's requests. The second segment suffers by vulnerabilities related to the intermediate servers. These vulnerabilities have to do mainly with the routing of the messages, as well as with the way the servers react in certain conditions. This segment obtains fewer rules, as limiting the freedom of the intermediate servers is not advised. The third segment corresponds to the communication that takes place within the callees domain.

The above categories may, at first glance, seem obvious. This is not the case, as in the SIP protocol every element that participates in a session does not have a particular role for the entire session. For example, an entity that participates to a SIP negotiation process might be also in the status of receiving/transmitting SIP messages (requests/ responses).

5 Development of Scenarios

Most attacks to an information system are realized through the exploitation of one or more of its vulnerabilities [12]. In the SPIT context, such a process could be based on, first, gathering a list of SIP URI addresses, then preparing the real SPIT message, and, finally, forwarding the SPIT message to the intended recipients. Such a series of steps could be proved useful for preventing and/or handling future attacks.

Attack graphs and attack trees are used for modeling attacks against an information system, a computer, or a network [13]. In existing publications, i.e., [14], such a SIP-oriented SPIT attack model was proposed. The structure of this model consists of three levels (from the most generic to the most detailed), namely: (a) the SPIT attack method (i.e. the series of steps of an attack), (b) the SIP-oriented SPIT attack graph (description of the attack method through the relationships among abstract attacks), and (c) the SIP-oriented SPIT attack trees (analysis of every abstract attack).

The model, which is depicted in Figure 4, describes the SPIT attack scenarios and provides a method for modeling them. In detail, the attack graph is presented in the left part. The basic nodes (1 to 7) represent the abstract attacks [12], namely: (1) find and collect users addresses, (2) send bulk messages, (3) proxies-in-the-middle attack, (4) maximize profit, (5) hide identity-track when setting-up an attack, (6) hide identity-track when sending a SPIT call/message, and (7) encapsulate SPIT in SIP messages. The arrows depict the possible connections/relations between the attacks.

The graph does not have a single start-node or end-node; it only demonstrates the exploitation of SIP protocol vulnerabilities. On the other hand, the right part of the figure presents the further analysis of one abstract node, i.e., it shows how node no. 7 is broken down in a more detailed attack tree.

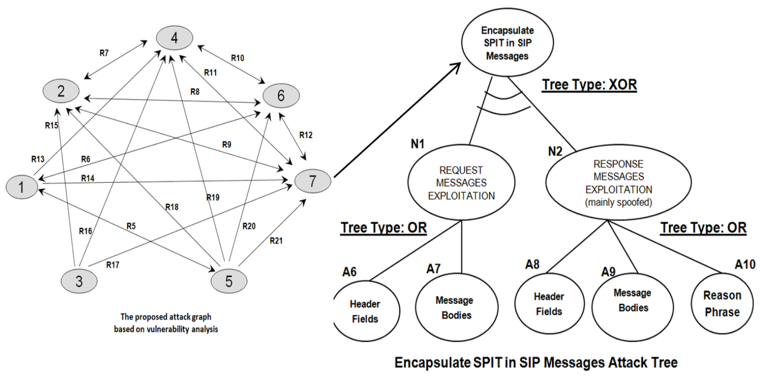
4
2
6
7
1
3
5
R5
R6
R7
R8
R9
R10
R11
R12
R13
R14
R15
R16
R17
R18
R19
R20
R21
The proposed attack graph based on vulnerability analysis
Encapsulate SPIT in SIP Messages
Tree Type: XOR
N1
REQUEST MESSAGES EXPLOITATION
N2
RESPONSE MESSAGES EXPLOITATION (mainly spoofed)
Tree Type: OR
Tree Type: OR
A6
Header Fields
A7
Message Bodies
A8
Header Fields
A9
Message Bodies
A10
Reason Phrase
Encapsulate SPIT in SIP Messages Attack Tree

Fig. 4. SPIT attack graph and an example of an attack tree

The model depicts how a spitter can exploit a SPIT-related vulnerability. The attack graph and attack trees can be transformed into attack scenarios (a number of example attack scenarios are described in Table 2) through the usage of an attack language like, SNORT attack language. Such a language can facilitate the transformation of a high level attack scenario into specific attack signatures (aka. definitions), which can be used can be used for detecting a SPIT attack.

Herein we will transform the attack scenarios into SPIT-oriented attack signatures. Then, the signatures will be encompassed by a SIP entity (mainly by SIP Proxy Server) for detecting SPIT. The XML language was selected for representing SPIT attacks, as its syntactical capabilities offer an adequate way for transforming high level attack scenarios into attack signatures.

6 Anti-SPIT Policy-Based Management

The proposed adaptive anti-SPIT Policy-based Management (aSPM) approach will be presented in this section. The approach tends to identify all SPIT attacks that are recognized by the attack scenarios and - at the same time - to react and respond according to the VoIP stakeholders wishes.

6.1 The Policy Condition Element

The main element of a policy description is a condition. A policy condition is a pattern of an identified SPIT attack scenario, as this is extracted by an attack graph. Such a condition is the key element for the detection of any possible SPIT attack. More specifically, a condition is formulated by extracting from an attack scenario specific characteristics and attributes that describe a SPIT attack. An example of an attack scenario description that can be used for this purpose is the following: *The callers user agent receives a response with a 200 message/code, which includes multiple addresses in Contact field and the value of the From field is equal to one of the values of the Contact*. According to it, the caller has more than one SIP addresses and introduces them in multiple contact fields. If these fields were inserted by a malicious user, then the next time the callee will try to communicate with the caller, she may call one of the alternative addresses. This leads to a possible redirection of the call towards an answering machine that plays pre-recorded SPIT messages. To deal with such an attack, we should identify these attributes of the scenario, which the SPIT detection will be based on. In this example, the appropriate attributes (or sub-conditions) are:

1. The message code is 200.
2. There are multiple Contact fields.
3. The equality between the value from the *From* header field and the value of the *Contact* header field.

The policy condition is the result of the logical aggregation of the three attributes, i.e.: *Condition=[Code=200* $\oplus$ *Contact:Multiple* $\oplus$ *From* $\approx$ *Contact].* The condition is defined, in general, as: Condition=f(c1,c2 ,,ck)=c1$\circ$ c2$\circ$ $\circ$ ck,

Table 2. Examples of attack scenarios[1]

Scenario	**Attack graph nodes and exploitation of vulnerabilities**
The spitter listens to the multicast for collecting users SIP URI addresses. Then, by using the contact header and the Alert-Info field of the INVITE message, forwards the SPIT message to the list of victims. Series of nodes: *Node 1 to Node 7*	The spitter starts the attack from node 1, namely *Find and Collect Users Addresses*. This is accomplished by exploiting the vulnerability *listening to a multicast address*. Then, the attacker goes to node 7, where the goal is to encapsulate *SPIT in SIP messages*. This is accomplished by the exploitation of the *SIP request messages* and especially the SIP headers fields *INVITE* and *ALERT-INFO* respectively.
The spitter exploits hijacked SIP proxies and sends bulk SPIT messages to an application that is using a multicast channel. Series of nodes: *Node 3 to Node 2*	The spitter starts from node 3 (*Proxies-In-The-Middle Attack*) and exploits Re-INVITES message header field vulnerability. Then, spitter transits to node 3 (*Sending Bulk Messages*), where she exploits the vulnerability of sending messages to the multicast address, that the application is using to provide content to multiple users (e.g. video conference).
The attacker sends ambiguous requests to proxies so as to collect SIP URIs addresses of her potential SPIT victims. Then by exploit the Response message and especially the From and Contact header fields forwards her SPIT message. Series of nodes: *Node 1 to Node 7*	The spitter starts the attack from node 1, namely *Find and Collect Users Addresses*. This is done by exploiting the vulnerability of *sending ambiguous requests to proxies*. Then, the attacker goes to node 7, where she uses a response message (*exploitation of response messages*) with a 200 message/code by *exploiting the Contact and From header fields* (i.e., includes multiple addresses in Contact field and the value of the From field is equal to one of the values of the Contact).

where ci is a suNote that footnotes associated with "floated objects" like tables or figures may have a problem insofar as the footnote might not follow the floated object.b-condition and $\circ$ denotes a logical operator to be $\oplus$ or $\otimes$ [2].

6.2 The Policy Action Element

The second element of a policy is the action (control, countermeasure). In SIP, proactive SPIT countermeasures can properly adjust the reaction of the negoti-

[1] The left column presents an abstract description of the attack scenario, while the right column depicts how each attack scenario is accomplished through the exploitation of specific nodes of our SPIT related attack graph

[2] The operators that are used in sub-conditions are: (1) = : equal, (2) **:** : times of header appearance (Multiple, One, None), (3) $\approx$: approximately equal, (4) ¿ : greater, and (5) ¡: less.

ation participating entities. The actions to be taken are divided in three main categories:

a. *Block:* It denotes the rejection of a SIP message. The action is enforced when we are sure that specific conditions are satisfied, therefore the message has been recognized as SPIT. The SIP action for this message is 403 *Forbidden"*.

b *Block with description:* It also refers to the rejection of a SIP message. The difference with the above category is that, in this case, a description of the reason why the message was rejected is sent to the request entity. This assists the caller or her domain to re-send the message, so as to meet the necessary requirements of the callee or her domain. A typical example of this action is a SIP 405 message with *Method not Allowed* description-information.

c *Notify:* It suggests that the SIP message is not rejected and will be forwarded, as the administrator/user desires. In this case, a notification is usually sent to the caller, and the message is redirected to an application that is responsible for supporting the communication. The caller usually receives a 183 SIP message with description *Session in Progress.*

6.3 From the Policy Elements to the Policy-Based Approach

An example of a condition and its corresponding actions appears on Table 3.

Table 3. A condition and suggested actions

Condition	**Code=200 $\oplus$ Contact:Multiple $\oplus$ From $\approx$ Contact**
List of possible actions	1.The UAC uses the specific address to compose upcoming messages
	2.The UAC renews the entries for the specific UAS
	3.User is informed for the new SIP addresses.
	4.The UAC rejects the call and returns a Message 403 (Forbidden)
	5.The UAC rejects the call and returns a message 606 (Not Acceptable)
	6.The UAC forwards SIP message to another entity and returns a message 183 (Session in Progress)

The policy element, together with its underlying components (condition element, actions element), are categorized on the basis of the communication segment in which each can be enforced. The entities that participate in each segment can have a different policy, i.e., the policy instance for each entity includes a different set of policy elements. Each anti-SPIT policy, and the corresponding policy elements, are defined and integrated manually by the administrator of each communication segment, presented in section 4. On the basis of the above, the proposed adaptive anti-SPIT policy-based management approach is depicted in Figure 5.

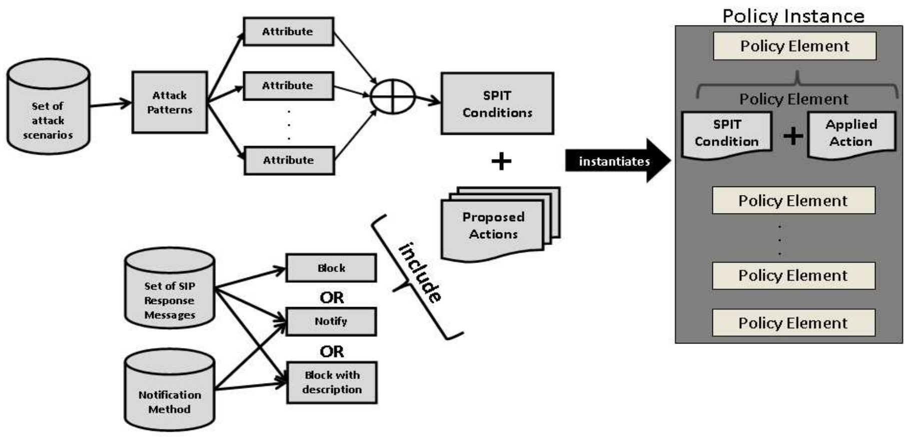

Fig. 5. aSPM: Adaptive anti-SPIT policy-based management

7 aSPM Proposed Architecture

As the applicability of the aSPM is an important part of its value, two steps were taken towards this direction. First, the anti-SPIT policy was described in a structured form, i.e., as an XML schema [15]. Second, the anti-SPIT policy was applied to an existing SIP infrastructure, i.e., the SIP Express Server (SER), an open source software product, currently use by organizations including Columbia University, Swiss Federal Institute of Technology, etc. [16,17].

7.1 XML Representation

XML is a markup language, which can represent data in a structured way. In our case, the XML schema[3] basically includes the identification characteristics of the attack attributes, together with the relation between them. The schema is developed in order to be: (a) easy for the administrator to develop a policy element, and (b) easily processed by an automated procedure. The main component/tags[4] of the schema will be described in the sequel, while the whole schema appears in the Appendix.

The main element of the XML policy structure is the *RuleItem*. A *RuleItem* consists of two elements, the *Subject*, on which the condition is applied, and the *Rule*, which obtains the policy element. The *subject* tag contributes in not having multiple policies for the same entity, as each communication entity takes a variety of roles during the SIP negotiation. In our case, the possible values of the *Subject* are: (a) Caller, (b) Callee, (c) Callers proxy, and (d) callees proxy.

The other element of the *RuleItem* is the *Rule*, which aims at identifying a certain condition and introducing the proper action, when the condition is met.

[3] XML Schema Definition (XSD).

[4] The remaining XML schema is presented in the Appendix.

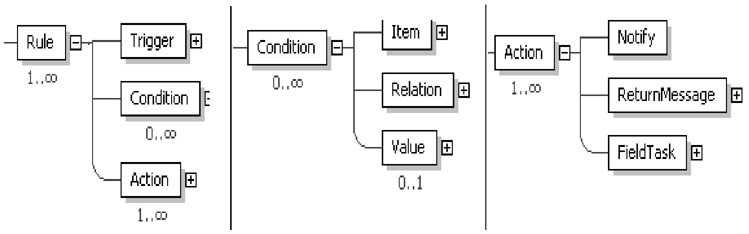

Fig. 6. Main XML elements

A *Rule* element consists of three tags, namely: (a) the *Trigger* tag, which denotes when the rule is checked, and its values are: Receive Message, Create Message, (b) the *condition* tag, and (c) the *Action* tag, which refers to the action that must be applied.

The condition can appear one, many, or no times in a Rule, in order to fulfill the produced attack pattern (i.e., sometimes there are no conditions to be fulfilled for an action to take place). This occurs when there are rules, which are mandatory for a specific event in a policy. These events are related to the Subject element and not to the condition element, which exists in the Rule.

The second tag of *Rule* Item, i.e., the *condition*, consists of: (a) the *Item* tag, on which the condition is checked, and it can be a header field or a request type (INVITE, OPTION, etc.), (b) the *Value* tag, which is the value of the *Item*, and (c) the *Relation* tag, which is the relation/logical operator between the *Item* and the proposed value. The *Relation* tag, defines whether the value of the *Item* should be equal to the *Value* Item or the exact difference from it. Also, the Relation element is used to indicate a property, of the *Item* element, like multiplicity or existence. The third part of the *Rule* tag is the *Action.* The action element is processed only if the *Trigger* and the *Condition* are fulfilled. An action consists of (a) *Notify*, which suggests the notification procedure that should be followed, (b) *Return Message*, which enforces the format and code of a new message to be send back to the appropriate entity, and (c) *FieldTask*, which contains all the actions that affect the header fields of a SIP message. Figure 6 depicts the main XML elements.

7.2 aSPM Integration

The aSPM Architecture In this section we describe how an anti-SPIT policy can be integrated in a SIP infrastructure. The proposed approach is shown in Figure 7. The approach is based on three basic elements, namely:

a) The SIP parser. It is an automated process, integrated to the SER server and used to support the routing of the incoming SIP messages. The SIP parser can scan SIP messages and extract the message attributes (the main attributes are the header fields, such as From, Contact, and SIP-URI and their values).

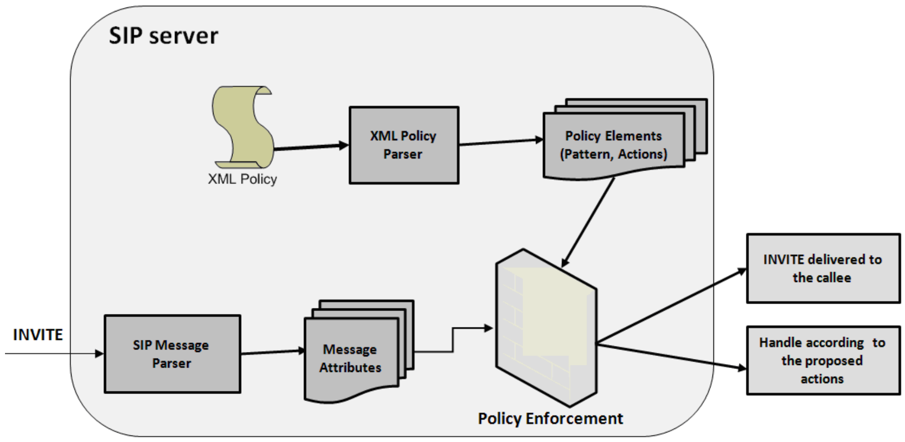
SIP server
XML Policy
XML Policy Parser
Policy Elements (Pattern, Actions)
INVITE
SIP Message Parser
Message Attributes
Policy Enforcement
INVITE delivered to the callee
Handle according to the proposed actions

Fig. 7. The aSPM Architecture

b) The XML parser. It is an interface that allows the navigation to the entire document, as if it was a tree of Nodes. The parser reads XML into memory, and provides easy access to tag values of the document.
c) The policy enforcement (or decision) point. The input to it is the parsed xml document, together with the message attributes. Two actions take place in this module: (a) all the conditions are checked, so as to find out which is fulfilled and which not (actually, first a SIP message is received and parsed, and then the message attributes are checked against the policy element), and (b) if one or more conditions are met, then the proposed action (described in the fulfilled policy element) does take place. If several conditions are met, then the stricter action is executed.

8 Brief Conclusions and Further Research Plans

SIP-based VoIP environments seem to gain a lot of attention, especially due their low cost for telephony services. However, the SPIT threat and its underlying SIP-related vulnerabilities pose a considerable concern that should be addressed.

In this paper we proposed an anti-SPIT policy-based management system (aSPM), with an eye towards the effective management of SPIT phenomenon. The suggested approach was primarily based on a SIP protocol threat and vulnerability analysis, which results in the identification of a series of attack scenarios. Then, the attack scenarios were analyzed, in an effort to define SPIT detection rules. These rules led to the identification and description of specific actions and controls, capable of countering and mitigating SPIT attacks. An XML schema was proposed as a means for both, first, describing the detection rules, and, second, stipulating the SPIT handling controls and actions. Finally, it was demonstrated how the aSPM approach could be practically integrated into a real SIP environment.

Regarding future plans, we aim to enhance the aSPM by supporting the integration of detection rules in a dynamic way, regarding for example user preferences and feedback [18]. Furthermore, we aim to analyze how a VoIP infrastructure, where aSPM is applied, can interoperate with other frameworks. In particular, we plan to check how the information resulted from the use of the aSPM in a given VoIP environment could be facilitated in different VoIP environments, with an eye towards evaluating its intra-VoIP environments application potential.

Acknowledgments. This work has been partially performed within the SPIDER (COOP-32720) Project, which is funded by the European Commission under FP6. The authors would like to thank our colleagues M. Theoharidou and Y. Mallios for their constructive comments and suggestions.

References

1. Rosenberg, J., et al.: Session Initiation Protocol (SIP), RFC 3261 (June 2002)
2. El Sawda, S., Urien, P.: SIP security attacks and solutions: A state-of-the-art review. In: Proc. of IEEE International Conference on Information and Communication Technologies: From Theory to Applications (ICTTA 2006), April 2006, vol. 2, pp. 3187–3191 (2006)
3. Rosenberg, J., Jennings, C.: The Session Initiation Protocol (SIP) and Spam, RFC 5039, Network Working Group (January 2008)
4. Tschofenig, H., Wing, D., Schulzrinne, H., Froment, T., Dawirs, G.: A document format for expressing authorization policies to tackle spam and unwanted communication for Internet Telephony (draft-tschofenig-sipping-spit-policy-02.txt)
5. Lennox, J., Wu, X., Schulzrinne, H.: Call Processing Language(CPL): A Language for User Control of Internet Telephony Services. RFC 3880, Columbia University (October 2004)
6. Quittek, J., et al.: Prevention of Spam over IP Telephony (SPIT). NEC Technical Journal 1(2), 114–119 (2006)
7. Sloman, M., Lupu, E.: Security and management policy specification. IEEE Network, Special Issue on Policy-Based Networking 16(2), 10–19 (2002)
8. Cisco Systems, Session Initiation Protocol gateway call flows and compliance information SIP messages and methods overview, `http://www.cisco.com/application/pdf/en/us/guest/products/ps4032/c2001/ccmigration09186a00800c4bb1.pdf`
9. Cisco Systems, SIP Messages and Methods Overview, `http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/rel_docs/sip_flo/preface.pdf`
10. Dritsas, S., Mallios, J., Theoharidou, M., Marias, G., Gritzalis, D.: Threat analysis of the Session Initiation Protocol regarding spam. In: Proc. of the 3rd IEEE International Workshop on Information Assurance (WIA 2007), April 2007, pp. 426–433. IEEE Press, USA (2007)
11. Marias, G.F., Dritsas, S., Theoharidou, M., Mallios, J., Gritzalis, D.: SIP vulnerabilities and antiSPIT mechanisms assessment. In: Proc. of the 16th IEEE International Conference on Computer Communications and Networks (ICCCN 2007), Hawaii, August 2007, pp. 597–604. IEEE Press, Los Alamitos (2007)

12. Mehta, V., Bartzis, C., Zhu, H., Clarke, E., Wing, J.: Ranking Attack Graphs. In: Proc. of Recent Advances in Intrusion Detection, September 2006, pp. 127–144. Springer, Germany (2006)
13. Schneier, B.: 'Attack trees', in Secrets & Lies: Digital Security in a Networked World, pp. 318–333. Wiley, Chichester (2000)
14. Mallios, Y., Dritsas, S., Tsoumas, S., Gritzalis, D.: Attack modeling of SIP-oriented SPIT. In: Proc. of the 2nd International Workshop on Critical Information Infrastructure Security (CRITIS 2007), October 2007, Springer, Spain (to appear, 2007)
15. Bertino, E., Castano, S., Ferrari, E.: On specifying security policies for web documents with an XML-based language. In: Proc. of the 6th ACM Symposium on Access Control Models and Technologies, pp. 57–65 (2001)
16. SIP Express Router (SER), Iptel.org, `www.iptel.org/ser`
17. Example SER deployments, `http://mit.edu/sip/sip.edu/deployments.shtml`
18. Guang-Yu, H., Ying-Youm, W., Hong, Z.: SPIT Detection and Prevention Method in VoIP Environment. In: The Third International Conference on Availability, Reliability and Security, pp. 473–478 (2008)

Appendix

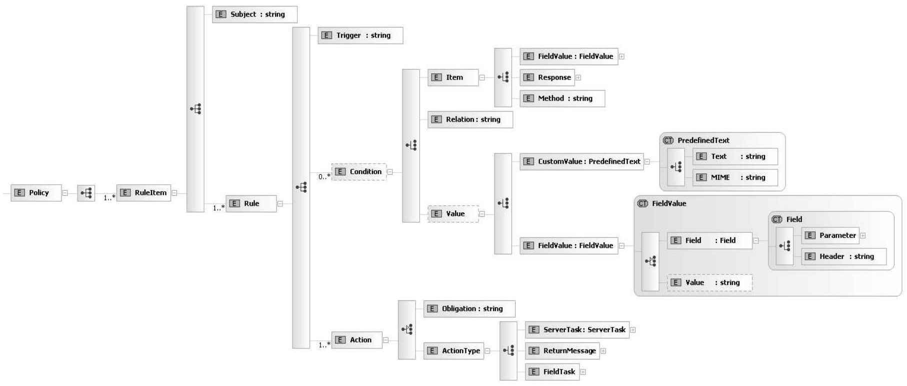

Fig. 8. Part of XML Schema