An Ontology-Driven antiSPIT Architecture

Stelios Dritsas and Dimitris Gritzalis

Information Security and Critical Infrastructure Protection Research Group
Dept. of Informatics, Athens University of Economics & Business (AUEB)
76 Patission Ave., Athens, GR-10434 Greece
{sdritsas,dgrit}@aueb.gr

Abstract. Over the last years Voice-over-IP (VoIP) is getting widespread adoption from business and residential customers.The preference towards VoIP services stems from the fact that VoIP provides many ways to communicate, with a lower cost than traditional telephony. However, VoIP in its present form may allow malicious users to exploit a number of vulnerabilities, make bulk unsolicited telephony calls and send bulk unsolicited instant messages. This exploitation is referred to as Spam over Internet Telephony (SPIT). In this paper we introduce an antiSPIT management architecture, by using ontologies, which allow domain administrators to detect and handle SPIT automatically, based on predefined requirements and preferences.

Keywords: Security, Availability, VoIP, SPIT, Ontology.

1 Introduction

The explosive growth of the Internet has introduced a wide array of new technological advances and more sophisticated end-user services. One of them is Voice-over-IP (VoIP), which offers telephony and other multimedia services over existing data networks. Some of the basic reasons that make Voice-over-IP (VoIP) increasingly appealing are: (a) VoIP's seamless integration with the existing IP networks and the Internet, (b) lower costs compared to PSTN telephony, (c) use of computer-based soft-phones, (d) existence and implementation of sophisticated end-user services in terms of portability, accessibility and convergence of telephone networks.

Currently, the majority of VoIP implementations are based on the Session Initiation Protocol (SIP), which tends to be the dominant protocol in VoIP environments [7]. Despite its benefits, SIP has a number of vulnerabilities which can negatively impact the use of VoIP services [1]. One introduced threat is the capability of malicious users to make bulk unsolicited telephony calls and/or send bulk unsolicited instant messages. This situation is a new form of spam, which - in the case of VoIP environments - is called Spam over Internet Telephony (SPIT) [2,3]. Although several anti-SPIT approaches have been proposed, their effectiveness has been characterized as inadequate due to their ad-hoc nature, their strong dependence to the used context, and to their inability to fully take into account the real-time nature of VoIP services [4,5]. This situation necessitates the development and deployment of specific anti-SPIT mechanisms and techniques in an effort to thwart SPIT phenomenon and enhance VoIP further pervasiveness.

A.B. Sideridis and Ch. Z. Patrikakis (Eds.): e-Democracy 2009, LNICST 26, pp. 189–198, 2010.

The effective management of SPIT phenomenon depends on many factors and on the specific characteristics that are introduced in a VoIP environment. Hence, in this paper we propose a SPIT management architecture that takes into consideration all the requirements posed so as to manage SPIT. The proposed architecture is based on: (a) the anti-SPIT policies that might be adopted in any SIP-based environment and (b) our ontology model (ontoSPIT), which operates as a policy enforcement tool.

The paper is organized as follows: First, we present some generic issues regarding SPIT phenomenon and the SPIT management procedure. In the sequel, we present our proposed SPIT management architecture in conjunction with its basic building blocks. Furthermore, in section 4 we demonstrate how our approach could be integrated in a SIP-based VoIP environment as well an example of an anti-SPIT policy rule and how this could be included in our ontoSPIT model. In section, 5 we evaluate our proposal in terms of (a) how they respond to some SPIT attacks. Finally, we conclude by providing the reader with some noteworthy remarks.

2 Spam over Internet Telephony (SPIT)

VoIP is an advancing technology that provides several benefits to its users. These benefits have been acknowledged, and VoIP users are continuously increasing over time. However as with any technological advancement, the rapid adoption of VoIP technologies will introduce new types of threats and will attract malicious users. Such a threat is SPIT, which is defined as the sending of a set of bulk unsolicited voice calls or instant messages [1].

Currently, three different types of VoIP spam forms have been recognized [2]: (a) Call SPIT, which is defined as bulk, unsolicited session initiation attempts in order to establish a multimedia session, (b) Instant Message SPIT, which is defined as bulk, unsolicited instant messages and it is well known as SPIM, and (c) Presence SPIT, which is defined as bulk, unsolicited presence requests so as the malicious user to become a member of the address book of a user or potentially of multiples users.

SPIT starts getting recognized as a possible serious problem in VoIP networks. The research community foresees that it will be very attractive for spammers and telemarketers (spitters) in the near future; a situation that may cause major annoyance to end-users. This interest of researchers for SPIT can be observed by the proposal of many anti-SPIT mechanisms and techniques. These mechanisms rely on well researched solutions and approaches for the email spam. However such approaches might not be satisfactory for VoIP, and some issues with these mechanisms have already been identified [4,5]. Hence, one can conclude that the SPIT management process (i.e. detection and handling of SPIT calls/messages) is not a simple one. It requires actions that take into account not only the specific characteristics of the VoIP technology but, also, the requirements posed by the administrators of each SIP-based VoIP domain. In Fig. 1 we provide a macroscopic view of the SPIT management. It is obvious that the prevention, detection and reaction of SPIT incidents, should be considered very carefully in an effort to manage SPIT phenomenon efficiently and effectively.

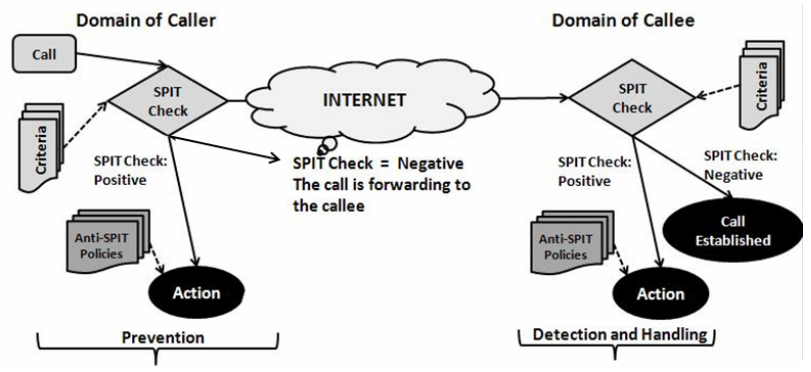

Fig. 1. SPIT management

3 A Proposed Anti-SPIT Architecture

In this section we present a SPIT management architecture. The architecture is based on two major components: a) anti-SPIT policies and b) the ontoSPIT model. We will first describe these two components, and then we will show how these components blend together in our architecture in order to manage SPIT.

3.1 Anti-SPIT Policies

Security policies can play a significant role in the enforcement of the basic security properties, namely: confidentiality, availability and integrity [8]. In this context, and with an eye towards to SPIT phenomenon handling, we define an anti-SPIT policy as the set of rules that should be adopted in an effort to detect and counter possible SPIT attacks and incidents.

In general, an anti-SPIT policy should include a set of rules that describe the actions to be taken for handling SPIT (i.e. actions that should be considered whenever a SPIT call/message is detected). The rules are consisted of two major components:

a) The underlying *conditions*, in charge of detecting possible SPIT voice calls and/or instant messages. The conditions are based on specific SPIT detection criteria that have been identified in [9], as well as on SPIT attack scenarios [10].
b) The appropriate *actions*, which represent the actions that should be adopted so as to handle an identified SPIT call and/or message. The actions that we have defined in the context of our proposed architecture are the following:

- *Allow:* In this case, the SIP message is forwarded to the next proxy or to the end-user.
- *Block:* This case 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.
- *Check Further*: This action is referred to the activation of further checks (e.g. CAPTCHA tests) so as to make more absolute conclusions regarding the nature of a SIP message (if it is SPIT or not).

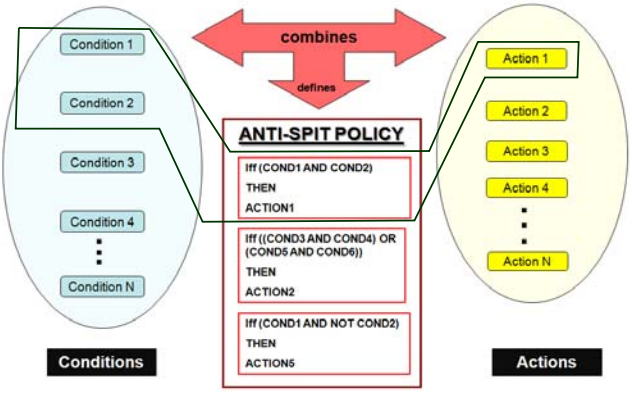

Fig. 2. Anti-SPIT policies representation

Fig. 2 depicts how each policy is represented by its underlying rules (i.e. the combination of conditions and actions).

3.2 The OntoSPIT Model

An ontology is an explicit specification of a conceptualization, which can be used to describe structurally heterogeneous information sources, helping both people and machines to communicate in a concise manner by supporting knowledge sharing and reuse [11,12]. The reusability approach is based on the following assumption: if a modeling scheme is explicitly specified and mutually agreed by the parties involved, then it is possible to share, reuse, and extend knowledge.

In this paper we adopt a SPIT management ontology model (ontoSPIT) that provides a SPIT-oriented knowledge base and enhances the sharing of its information and components [13]. By using ontologies, we can use their advantages in an effort to help stakeholders of SPIT domain (i.e. administrators and end-users) make decisions regarding the efficient SPIT management. These decisions should be based on pre-defined assumptions or on tacit knowledge entailed by the amount of SPIT related information generated by different sources, such as anti-SPIT systems, VoIP domains, SPIT related SIP vulnerabilities, SPIT attack patterns and scenarios, etc.

3.3 SPIT Management Architecture

Our proposed SPIT management architecture tries to handle SPIT by incorporating a series of SPIT-related information including the SPIT related vulnerabilities of the SIP protocol [14], SPIT-related attack patterns and scenarios [10], and specific SPIT incident detection criteria [9]. This is essential information that provides a satisfactory starting point for countering SPIT phenomenon at its early stages. The following picture depicts our proposed architecture, while in the next paragraphs we describe the steps needed to adopt our architecture in a VoIP domain.

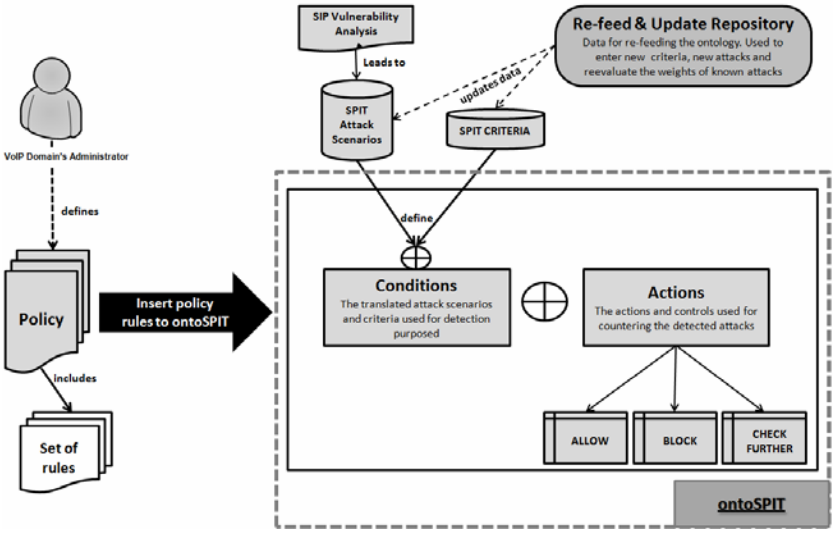

Fig. 3. The proposed anti-SPIT architecture

Step 1: *Anti-SPIT Policy Definition:* The first step is to define the anti-SPIT policies that will be adopted in the SIP-based VoIP domain. The administrator of the domain defines the rules of the policy, i.e. the conditions under which a SIP message is checked and the desired actions that will be taken if a SPIT suspicious message has been identified.

Step 2: *Anti-SPIT Policy Adoption:* After the definition of the policy, it is important to translate it into a format that is compatible with the ontoSPIT model; that is the administrator of the VoIP domain needs to express the policy rules into the desired conditions and actions that will be included into the ontology in a language that will be recognizable from the ontoSPIT model. One such language is the OWL language [15,16,17], which was also used in the original OntoSPIT model [13].

Step 3: *Anti-SPIT Policy Enforcement:* The final step is to enforce the defined anti-SPIT policy. This is taken care by the implementation and execution of the ontoSPIT model. When a SIP message is received the ontology is instantiated and its basic concepts and classes take specific values. Based on these values every SIP message is handled according to the conditions and actions that were defined by the policy. Therefore, the ontoSPIT model is responsible to draw conclusions about the nature of each SIP message (i.e. whether it is a SPIT message or not) by fulfilling the requirements posed by the underlying anti-SPIT policy.

4 Use of the Architecture

In this section we show how the proposed architecture could be integrated in a SIP-based VoIP environment, which is based on the SIP Express Server (SER), an open source software product that provides the basic SIP services [19,20]. A macroscopic view of our approach is presented in Fig. 4.

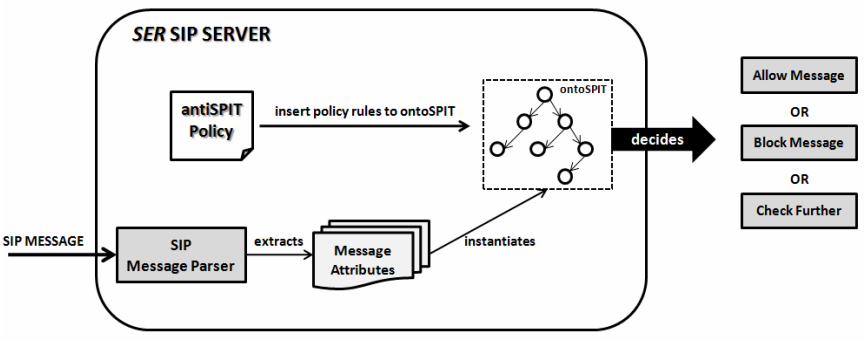

Fig. 4. Use of the proposed anti-SPIT Architecture

The basic elements of our approach are the following:

a) *SIP Message Parser:* This module is an automated process, integrated to the SER server and is used to support the routing of the incoming SIP messages. The SIP parser can scan SIP messages and extract the message attributes that will be used to check whether the messages are SPIT or not.
b) *antiSPIT Policy:* This element refers to the antiSPIT policy that is adopted by the specific SIP-based VoIP domain, which is expressed using the SWRL language [18], so as to be incorporated into the ontoSPIT model.
c) *ontoSPIT:* This module is the implemented ontoSPIT model which is responsible for enforcing the above-mentioned antiSPIT policy. The ontoSPIT model has been implemented using the well-known Protégé tool [21].

Having in mind the aforementioned approach we now present an example of a policy rule and how this rule is incorporated into ontoSPIT model using the SWRL language. The rule is responsible for checking the PRIORITY header of a SIP INVITE message and if specific conditions are met, then the message is characterized as SPIT and hence is blocked.

The condition of the rule checks if the SIP message has the PRIORITY field set as urgent and at the same time if the FROM header includes a sender that is not trusted (i.e. not presented in the white list of the domain and/or receiver of the message). The action of the rule defines that if the abovementioned condition is true, then the message is blocked. The overall rule, as well as how it is written using SWRL language, is depicted in Table 1.

5 Architecture Evaluation

In section 1 we mentioned that the current proposed anti-SPIT techniques and mechanisms are influenced by the methods used for countering spam. Therefore, they fail to take into account the SPIT vulnerabilities that are introduced by the use of SIP protocol, even when some of the mechanisms focus on SIP to handle SPIT.

Having in mind the list of SPIT oriented SIP vulnerabilities as presented in [14], we provide an evaluation of the proposed architecture using as criteria the SPIT related attacks that could be conducted in a SIP-based VoIP domain [5].

Table 1. Incorporating a policy rule into the ontoSPIT model

Rule for checking the PRIORITY header of a SIP Message	
IF (*SIP Message arrived*) **AND** ((<SIP Header=*PRIORITY*>) **AND** **IF** (< *PRIORITY* field = urgent>)) **AND** (**IF** (<*FROM* field != ID presented in WhiteList>))) **THEN** **BLOCK**	**Rule written in SWRL language:** SIPMessage(?m) ∧ ((has_header(?m,?h) ∧ has_field(?h,?f) ∧ Priority(?f) ∧ swrlb:equal (?f,"Urgent")) ∧ (has_field(?h,?f2) ∧ From(?f2)) ∧ Whitelist(?wl) ∧ contains(?wl,?x) ∧ differentFrom(?f2,?x)) ∧ (checking_of(?c,?m) ∧ previous_to_decision_1(?d,?c)))→ Block(?d)
Detailed SWRL Code	**Comments – Description**
SIPMessage(?m) ∧ (	**Whenever** a message m is received,
(has_header(?m,?h) ∧ has_field(?h,?f1) ∧ Priority(?f1) ∧ swrlb:equal (?f1,"Urgent")) ∧	**and** this message has a priority field with value "Urgent" (which means that the message m has a header h **and** this header h has a header-field f1 **and** this field f1 is of type Priority equal to "Urgent")
(has_field(?h,?f2) ∧ From(?f2)) ∧	**and** this header h has another field f2 of type From
(Whitelist(?wl) ∧ contains(?wl,?x) ∧ differentFrom(?f2,?x)) ∧	**and** there is a whitelist wl that contains the values x **and** the from field f2 is different from the values x
(checking_of(?c,?m) ∧ previous_to_decision_1(?d, ?c)))	**and** the checking c of this message m is previous to the decision d based on this checking c
→ Block(?d)	**then** the decision d must be of type Block

Moreover, we compare our approach with the current anti-SPIT mechanisms, so as to better illustrate its advantages in terms of a holistic SPIT management process. Table 2 presents our findings.

Table 2. Evaluation of the proposed architecture

Anti-SPIT frameworks / **SPIT attacks**	SPIT Prevention using Anonymous Verifying Authorities	SPIT Mitigation through a Network Layer Anti-SPIT Entity	SPIT Detection based on Reputation and Charging techniques	DAPES	Progressive Multi Gray-Leveling	Biometric Framework for SPIT Prevention	RFC4474	SIP SAML	DSIP	Voice SPAM Detector	Hidden Turing Tests	VoIP SEAL	**Proposed architecture**
Address harvesting													√
Multiple Account Instantiation													√
Open relays and proxies		√	√	√	√				√	√		√	√
Anonymity Services		√	√	√	√				√	√		√	√
Obfuscating message content													√
Support services		√	√	√	√				√	√	√	√	√
Sending messages to multicast addresses													√
Exploitation of forking proxies													√
Exploitation of registrars servers						√	√	√					√
Exploitation of re-INVITE request messages													√
Exploitation of the record-route header field													√
Exploitation of messages and header fields structure													√

6 Conclusions and Further Research

VoIP technology is gaining a recognizable market share. For this reason, significant concerns are raised regarding the possibility of an explosion for SPIT attacks to the extent of email SPAM. Recently some anti-SPIT mechanisms were proposed, but they neither took into account any SPIT-related threat or vulnerability analysis, nor any attack schemes that spitters may use.

In this paper, we proposed a SPIT management architecture based on specific anti-SPIT policies that might be adopted in any SIP-based VoIP domain. The enforcement of the policies is guaranteed through the use of a SPIT oriented ontology model (onto-SPIT). Our approach takes into consideration the SPIT-related SIP vulnerabilities and threats with an eye towards providing a holistic SPIT management framework.

Regarding future work, we aim to enhance our architecture, so as to take into consideration specific statistics data, regarding SPIT traffic, and to handle this traffic in a dynamic manner. Furthermore, we are looking to incorporate the ontology in general security ontologies, so as to enhance the existing models and thus also incorporate SPAM/SPIT management processes.

Acknowledgements. We would like to thank our colleagues Yannis Mallios (Carnegie Mellon University, USA) and Marianthi Theoharidou (Athens University of Economics & Business, Greece) for their helpful comments and suggestions.

References

1. VOIPSA, VoIP Security and Privacy Threat Taxonomy (October 2005), http://www.voipsa.org/Activities/taxonomy.php
2. Rosenberg, J., Jennings, C.: The Session Initiation Protocol (SIP) and Spam, draft-ietf-sipping-SPAM-03 (October 2006)
3. Dritsas, S., Mallios, J., Theoharidou, M., Marias, G., Gritzalis, D.: Threat analysis of the Session Initiation Protocol regarding spam. In: Proc. of the 3rdIEEE International Workshop on Information Assurance (WIA 2007), April 2007, pp. 426–433. IEEE Press, USA (2007)
4. Marias, G., Dritsas, S., Theoharidou, M., Mallios, J., Gritzalis, D.: SIP vulnerabilities and anti-SPIT mechanisms assessment. In: Proc. of the 16th IEEE International Conference on Computer Communications and Networks (ICCCN 2007), August 2007, pp. 597–604. IEEE Press, Los Alamitos (2007)
5. Gritzalis, D., Mallios, Y.: A SIP-based SPIT management framework. Computers & Security 27(5-6), 136–153 (2008)
6. Rosenberg, J., et al.: SIP: Session Initiation Protocol, RFC 3261 (June 2002)
7. Johnston, A.: SIP: Understanding the Session Initiation Protocol. Artech House (2004)
8. Sloman, M., Lupu, E.: Security and management policy specification. IEEE Network, Special Issue on Policy-Based Networking 16(2), 10–19 (2002)
9. Dritsas, S., Soupionis, J., Theoharidou, M., Mallios, J., Gritzalis, D.: SPIT Identification Criteria Implementations: Effectiveness and Lessons Learned. In: Samarati, P., et al. (eds.) Proc. of the 23rd International Information Security Conference (SEC 2008), September 2008, pp. 381–395. Springer, Milan (2008)

10. Mallios, J., Dritsas, S., Tsoumas, B., Gritzalis, D.: Attack modeling of SIP-oriented SPIT. In: Lopez, J., Hämmerli, B.M. (eds.) CRITIS 2007. LNCS, vol. 5141. Springer, Heidelberg (2008)
11. Gruber, T.: Toward principles for the design of ontologies used for knowledge sharing. In: Formal Ontology in Conceptual Analysis and Knowledge Representation, March 1993. Kluwer Academic Publishers, Dordrecht (1993)
12. Guarino, N.: Understanding, Building, and Using Ontologies: A commentary to "Using explicit ontologies in KBS development. International Journal of Human and Computer Studies 46(3/4), 293–310 (1997)
13. Dritsas, S., Dritsou, V., Tsoumas, B., Constantopoulos, P., Gritzalis, D.: OntoSPIT: SPIT management through ontologies. Computer Communications (April 2008) (in press)
14. 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, April 2007, pp. 426–433. IEEE Press, New Orleans (2007)
15. W3C Recommendation, The Ontology Web Language
16. OWL. W3C Recommendation. The Ontology Web Language (2004)
17. W3C. W3C Recommendation (10-02-2004), OWL Guide (2004)
18. Horrocks, I., Patel-Schneider, P., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A Semantic Web Rule Language Combining OWL and RuleML, The DARPA Agent Markup Language Homepage
19. SIP Express Router (SER), Iptel.org.
20. Example SER deployments, `http://mit.edu/sip/sip.edu/deployments.shtml`
21. Protégé, Ontology development environment (2005), `http://protege.stanford.edu/`