An Ontology Based-Policy for Deploying Secure SIP- based VoIP Services

Dimitris Geneiatakis, Costas Lambrinoudakis and Georgios Kambourakis

Laboratory of Information and Communication Systems Security
Department of Information and Communication Systems Engineering
University of the Aegean, Karlovassi, GR-83200 Samos, Greece
Tel:+30-22730-82247
Fax: +30-22730-82009
Email:{dgen, clam,gkamb}@aegean.gr

***Abstract* — Voice services over Internet Protocol (VoIP) are nowadays much promoted by telecommunication and Internet service providers. However, the utilization of open networks, like the Internet, raises several security issues that must be accounted for. On top of that, there are new sophisticated attacks against VoIP infrastructures that capitalize on vulnerabilities of the protocols employed for the establishment of a VoIP session (for example the Session Initiation Protocol - SIP).**

This paper provides a categorization of potential attacks against VoIP services, followed by specific security recommendations and guidelines for protecting the underlying infrastructure from these attacks and thus ensuring the provision of robust and secure services. In order to utilize (share) the aforementioned security guidelines and recommendations into different domains, it is necessary to have them represented in some formal way. To this end, ontologies have been used for representing the proposed guidelines and recommendations in the form of a unified security policy for VoIP infrastructures. This ontology-based policy has been then transformed to a First Order Logic (FOL) formal representation.

The proposed ontology-based security policy can be applied in a real VoIP environment for detecting attacks against a SIP based service, but it can be also utilized for security testing purposes and vulnerabilities identification.

The work presented in this paper has been focused to the SIP protocol. However, generalization to other signaling protocols is possible.

***Keywords*—SIP, VoIP, Ontology, Security Policies, Attack Description, Formalization**

I. INTRODUCTION

The continuously increasing convergence of data and voice networks drives telecommunication and Internet Service Providers (ISPs) to offer Voice – Telephony services over Internet Protocol (VoIP). Among the challenges that VoIP providers have to deal with is the attainment of availability levels that are at least equivalent to those of Public Switch Telephone Network (PSTN). Beyond network failures, it is true that VoIP availability is not only affected by security flaws stemming from the Internet architecture but also from new sophisticated attacks against, or vulnerabilities of, VoIP protocols, mainly during the establishment of a session [1],[2]. Moreover, the utilization of an open network, like the Internet, constitutes any VoIP infrastructure an easy target since there are many alternative methods and tools that can be employed by an attacker for launching an attack. On the other hand, PSTN does not suffer from similar problems since it is based on a closed network architecture.

Consequently, VoIP providers should seriously take into account the security issues arising at different levels of Internet architecture, in order to offer secure, reliable and robust services. Currently, most research work [3]-[5] is concentrated on general recommendations and guidelines for securing VoIP infrastructures. However, to the best of our knowledge there is no research work focusing on the provision of specific guidelines, best practices and policies for tackling or mitigating specific VoIP security flaws presented in [6],[1],[2]. At the same time the lack of a common attack description framework does not allow VoIP providers to effectively cooperate with each other in order to detect or / and repel an attack.

This paper contributes towards that direction by presenting, analyzing and formalizing guidelines and best practices that can be adopted during all deployment phases of a VoIP service. The result will be improved security, reliability and availability of the service. Our work mainly deals with the Session Initiation Protocol (SIP) [7] as it is the predominant signaling protocol for Next Generation Networks (NGNs). Even though the proposed guidelines are formalized through the employment of an ontology describing a policy based on SIP's security flaws [8] - aiming to provide such guidelines as a real service to SIP-based VoIP providers - it should be stressed that a similar description with only few modifications can be applied to alternative signaling protocols. Furthermore, VoIP providers could utilize the ontology representation not only for describing defense policies but also for testing the security robustness of their infrastructure.

The rest of the paper is structured as follows. Section II introduces background information regarding VoIP architecture and SIP protocol. Section III focuses on known vulnerabilities of SIP-based VoIP services. The possibility to trigger similar attacks against VoIP infrastructures that employ

other signaling protocols is also investigated in this section. Section IV introduces the guidelines that should be considered during the deployment phase of any VoIP service in order to successfully mitigate potential attacks. Section V demonstrates how ontologies can be used for representing the proposed guidelines as well as for transforming the ontology-based policy to a FOL formal representation. Section VI demonstrates how the proposed ontology based policy can be employed in a real environment, while section VII concludes the paper and provides some pointers for future work.

II. BACKGROUND

A. VoIP-Internet Architecture

A VoIP infrastructure inherits and utilizes various protocols from the Internet stack architecture. Specifically, at network and transport levels it employs the Internet Protocol (IP) [9] and TCP [10], UDP [11] or SCTP [12] protocols respectively. In addition, at the application level it does not only exploits well known protocols like DNS [13], DHCP [14] etc, but also dedicated ones that are used to handle sessions and transport media data. As illustrated in Figure 1, application level's protocols can be classified into the following categories:

- *Signaling*: Signaling Protocols are employed to handle a voice or multimedia session among two or more VoIP network entities. The most well known signaling protocols include: SIP [7], H.323 [15], MGCP [16], SKINY [17].
- *Utilities*: Utilities protocols like DNS and DHCP are used to offer additional services, such as address name resolution, dynamic configuration and many more.
- *Media*: Media protocols like RTP [18], SRTP [19] and ZRTP [20] are utilized to transmit media data (voice, video) among the entities that have previously established a session.

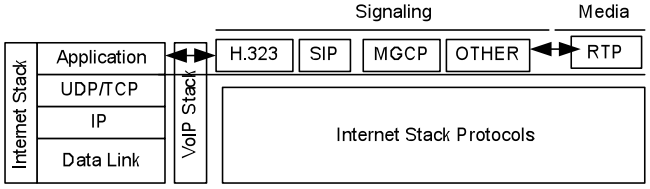

Figure 1. VoIP Protocol Stack

B. An overview of the Session Initiation Protocol

As already stated in Section II.A, SIP [7] is an application layer signaling protocol that inherits the HTTP message structure (see Figure 2) for handling (establishing, canceling, terminating) multimedia sessions over Internet among two or more participants. More specifically, whenever a SIP client (caller) wishes to establish a multimedia session with another SIP client (callee) generates a SIP INVITE request message and sends it to the appropriate server, which locates the callee and forwards the request to him.

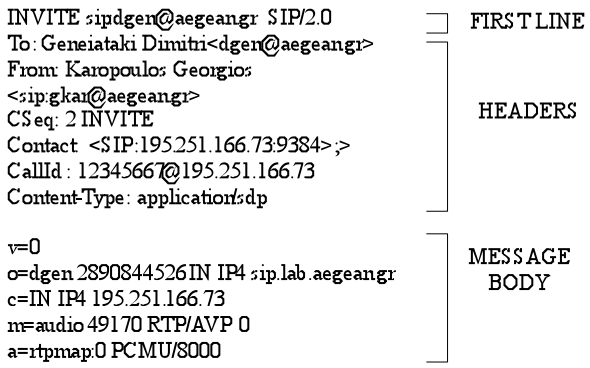

Figure 2. SIP INVITE Request message

If the callee accepts the call, generates and sends back the appropriate response, that is a 200 OK message, and the session is successfully established. The aforementioned procedure is illustrated in Figure 3. Without loss of generality, similar procedures for handling a session are implemented by other signaling protocols, like H323.

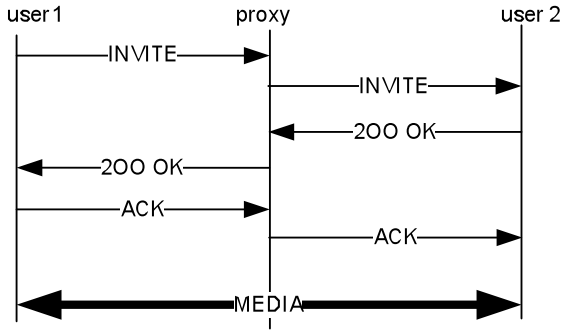

Figure 3. Session Establishment Procedure in SIP

III. ATTACKS-THREAT ANALYSIS IN SIP BASED VOIP SERVICE

At present, several researchers [1],[2],[6],[21] have highlighted various security flaws and vulnerabilities in VoIP services. Even though most of them focus on SIP, as it is the predominant protocol for NGN, similar attacks could be launched against VoIP infrastructures employing alternative signaling protocols. According to [1],[2] SIP based VoIP services are vulnerable to the following general threat and/or attack categories:

- Eavesdropping
- Parsing related
- Application level

A. Eavesdropping Attacks

The fact that gaining access to a communication channel is rather straightforward, in conjunction with the lack of effective confidentiality protection mechanisms in most VoIP systems, constitutes VoIP communication vulnerable to illegal and unauthorized monitoring for both signaling and media data. Furthermore, the fact that eavesdropping tools, like Ethereal (www.ethereal.com), are widely available and that

SIP messages are text based, makes such an attack very easy to accomplish. For instance, during an eavesdropping attack the attacker could capture a SIP message (as shown in Figure 2) and thus become aware of all session information regarding the caller and the callee.

Moreover, taking into account the fact that eavesdropping is always applied at the very early stages of other VoIP attacks (e.g. signaling as described further down in Section III.C.1) it should be considered as a serious flaw in SIP based VoIP telephony systems. It is thus clear that eavesdropping not only violates privacy but also (indirectly) affects communication reliability too.

B. Parsing Attacks

Parsers are considered as the core component of any communication system because they are in charge of the initial processing of all incoming or outgoing message. Hence, any instability in them can severely affect the availability of the provided service. However, there are several parser implementations that are not fully compliant with the underlying standards. Specifically, most parsers are only able to process well-formed messages, as that of Figure 2, without having any defense strategy against messages that do not conform to the corresponding specifications.

For instance, the PROTOS test suite [22] has been recently utilized in order to test existing SIP parsers' robustness while processing various different types of malformed messages. The corresponding results have demonstrated that the operation of most SIP parsers becomes unstable when processing malformed messages. Furthermore, there are cases where a perfectly valid SIP message has been crafted to hamper proper parsing [2]. Therefore, an efficient parser should be able to proactively discard any malformed messages in an attempt to keep the availability of the VoIP service high.

C. Application Level Attacks

Whilst an incoming or outgoing SIP message may be perfectly valid and thus go through any detection mechanism introduced during the parsing phase, as recommended in [23], it might target to breed an application level attack. For instance, the aggressor would possibly craft such a message in order to illegally terminate a session (signaling attack) or alternatively generate a bogus request in order to cause a DoS to the provided service making it unavailable to legitimate users (flooding attack).

1) Signaling Attacks

As signaling attack is defined an attempt to terminate or illegally modify an established session by creating a spoofed SIP message (like SIP BYE, SIP CANCEL, SIP UPDATE etc). In order to launch such an attack in a SIP realm the aggressor should learn the exact session's parameters. These parameters are included in the signaling messages exchanged prior to the establishment of the connection and could be eavesdropped as mentioned earlier in Section III.A. Upon that, the attacker generates the corresponding message (e.g. SIP BYE) and sends it to the appropriate SIP server in order to cause an illegal termination-alteration of the session. For instance, consider the case where a malicious user sends a SIP CANCEL message on behalf of the caller before the latter sends the SIP ACK message as illustrated in Figure 3. The result of this action is that the session in progress will be terminated illegally. It should be mentioned that the main reason that a malicious user is able to launch such an attack is the lack of appropriate authentication mechanisms. More details about signaling attacks can be found in [1] and [23].

2) Flooding Attacks

Resource consumption attacks, like flooding, are included among the most severe threats in any Internet application. According to this type of attack the malicious user tries to consume either system's resources or the available bandwidth, in an attempt to cause a DoS or substantially decrease service reliability and availability levels. An example of such an attack on Internet hosts is well known as Reflection Distributed DoS (RDDoS) [24]. Similar attacks can be also launched against any other type of Internet based service and the more critical the service is (e.g. VoIP) the more attention it gains for a resource consumption attack.

Several different types of flooding attacks that could be launched against SIP services are discussed in [2] and [25]. For example, a malicious user may try to cause a DoS to the registration service by sending bogus requests of SIP REGISTER messages, forcing the registrar to execute 'expensive' cryptographic operations. On the other hand, the attacker could focus to the calling service by generating several SIP INVITE requests and sending them to the corresponding SIP server in order to cause a DoS.

D. The Universality of VoIP Attacks and Threats

Although the attack categories presented in the previous subsections mainly concentrate on SIP based VoIP services, it is true that similar attacks / threats are also applicable to VoIP services employing alternative signaling protocols. In fact, eavesdropping techniques can be exploited against alternative signaling protocols by utilizing either tools like ethereal or proprietary sniffing modules able to specifically recognize these signaling protocols, as in the case of UNIStim Decoder provided by Nortel (www.nortel.com). Moreover, likewise to SIP parsing attacks an attacker may craft malformed messages in order to cause a DoS to the provided VoIP service; such types of attacks have been already published for H.323 [26]. Additionally, regarding SIP application level attacks, it is clear that the attacker can consume any VoIP system resource (independently from the signaling protocol employed) by generating a vast number of bogus requests against the corresponding server. As far as signaling attacks are concerned, it should be mentioned that as long as the attacker has the ability to eavesdrop session's parameters, he can launch a signaling attack similar to those presented for SIP systems.

Summarizing, threats and attacks against VoIP services are independent from the signaling protocol employed (see Table 1). Besides, the interconnection among different VoIP providers offers more opportunities to attackers while conveying attacks and their consequences from one domain to

another.

Threat –Attack	Affects	Applicable to other Signaling Protocols
Eavesdropping	Privacy	Yes
Parsing	Service availability-reliability	Yes
Signaling	Session availability	Likely
Flooding	Service availability-reliability	Yes

Table 1. Attacks and Threats in VoIP systems

IV. Security Guidelines and Recommendations for Employing Secure VoIP Services

In order to provide the appropriate security measures and guidelines for protecting the underlying infrastructure of the provided services, one should understand the root causes of the aforementioned threats/attacks. Generally speaking, independently from the service at danger, attackers always exploit a specific vulnerability or a combination of them in order to launch an attack. Table 2 provides the correlation among threats and vulnerabilities in a VoIP domain. Therefore, VoIP providers should take into serious consideration existing vulnerabilities in order to minimize the chances of suffering an attack.

Threat/Attack	Vulnerability
Eavesdropping	Lack of Confidentiality Mechanisms. Easy access to the communication medium
Parser	Implementation Errors
Signaling	Inadequate authentication mechanisms – Authentication flaws
Flooding	Inherent Internet flaws

Table 2. Correlation of Threats/Attacks and Vulnerabilities in VoIP

More specifically, any eavesdropping attempt could be identified by monitoring whether an unauthorized network component acts in promiscuous mode. In most cases, such an attack is considered as passive and thus it is very difficult to detect. Moreover, such incidents do not directly affect VoIP service availability, resulting in a low priority classification of the countermeasure against them. At this point it should be stressed that the employment of an end-to-end confidentiality mechanism seems infeasible due to the fact that some parts of the signaling data must be in cleartext for intermediate nodes in order to route the message to its final destination.

On the downside, all the remaining attacks, i.e. parsing, signaling and flooding, directly and substantially affect VoIP service availability and reliability. As a result VoIP providers must introduce the appropriate policies and security mechanisms in order to protect the services and increase their availability and reliability levels.

To start with, parsers should be able to identify if a message conforms to the syntax of the protocol. If this is the case the process should continue, otherwise the specific message must be discarded. To the best of our knowledge the only solution against malformed message attacks is proposed in [8]. In addition, to defend against signaling attacks and illegal message modifications, VoIP providers should employ appropriate message authentication and integrity protection mechanisms (similar to the ones presented in [27],[28]) both for request and response messages. Last but not least, providers should employ or develop proper flooding detection and prevention mechanisms for proactively identifying such attacks and reactively avoid resource consumption. Although flooding attacks as mentioned previously are considered one of the most severe threats, only few solutions have been proposed to protect VoIP systems [25],[29],[30]. Table 3 summarizes the aforementioned guidelines and recommendations; it is stressed that the proposed solutions are focused on mechanisms for SIP based VoIP services.

Security Guidelines - Recommendations		*Brief Description*	*Proposed Solution*	*Necessity*
G1	Detect unauthorized network components operating in promiscuous mode.	Detect network components trying to illegally eavesdrop on VoIP messages	Enable detection of promiscuous mode	Should
G2	Message validation according to the protocol grammar.	Protects parsers against malicious messages processing	[8]	Must
G3	Employment of appropriate message integrity and authentication mechanisms	Protects VoIP services against signaling attacks and illegal message modification	[27][28]	Must
G4	Employment of flooding detection and prevention mechanisms	Protects VoIP services against attacks targeting on resource consumption	[25][29][30]	Must

Table 3. General guidelines against VoIP security flaws

V. Utilization of Ontologies for the Formalization of Security Guidelines and Recommendations

Security guidelines and recommendations should be described in a uniform and formal way in order to share and introduce the same semantics among different domains. Currently, there are various languages (mainly implemented as part of Intrusion Detection Systems – IDSs) that describe attacks or/and security flaws for specific systems. However, such formalization is not easy to apply in heterogeneous architectures. For example, an attack represented in the attack language A, say STATL [31], cannot be utilized by an IDS like SNORT [32]. Furthermore, their semantics often lack of formal logic, while the same description cannot be employed for security testing purposes. Consequently, a uniform formal description of security guidelines and recommendations is necessary not only for detecting attacks against various architectures, but also for discovering vulnerabilities. Such formalization shall provide a complete and robust framework for the description of security policies in VoIP services.

According to [33] and [8] the required type of formalization can be achieved through ontologies. An ontology provides a

common understanding of the concepts within a specific domain and the relations between them, while the mapping in First Order Logic (FOL) can provide a formal description of the security issues for VoIP systems. Furthermore, the FOL mapping restricts the allowable interpretations of the non-logical symbols (e.g. relations, functions etc) and thus enables operations on different ontology instances using sound and complete theorem provers like Racer [34].

In order to detect intrusions and offer robust services over VoIP distributed architectures, it is necessary to share the security information among various entities. Towards this direction, ontologies can be utilized for representing the security guidelines and recommendations provided in Section IV. Such representations can also support the reuse of the same description; for instance, the same description could be employed by various applications to offer:

- Security guidelines as a real service.
- Security tests to check system robustness.
- An identification tool that examines VoIP logs for identifying security flaws.

A. The Ontology Representation

Figure 4 depicts the ontology pattern for security flaws of SIP based VoIP services that correspond to Table 3 guidelines G2, G3 and G4. The ontology can be seen as two main sub-ontologies: The *SIP-Message* and the *SIP-Attack* sub-ontologies. The former is directly related to the SIP-Message, which is considered the core component of any SIP-attack. The high level representation of the proposed ontology (listed in the Appendix I) has been based on DAML+OIL [35] language. Note that although guideline G1 may directly or indirectly violate VoIP security goals, it is not considered as a VoIP-oriented security issue but rather as a general one. Therefore it is not included in this representation.

1) The SIP-Message sub-ontology

The *SIP-Message* part of the ontology covers guideline *G2* (see also Part A of appendix I). It employs a specific message validation mechanism based on the SIP protocol grammar, as described in the RFC 3261 [7], providing explicit rules for determining whether a SIP message should be processed or not. More specifically, the *SIP-Message* sub-ontology comprises of the following classes:

- *First Line*: Represents the SIP request/response generated by a SIP agent in order to request a service or respond to a specific request. Every First-Line is composed by the Request-Response followed by the address, i.e. the Uniform Resource Identifier (URI), of the requested resource.
- *Header*: Represents the mandatory headers of every SIP message, providing also a description of their grammar.
- *Authenticate*: Represents whether a specific SIP-Message requires authentication or not.
- *Event*: Represents the events triggered by a SIP-Message, which may result in the transition of SIP system's state to some other one.
- *Time*: Represents the time at which a SIP-Message has been initially processed.

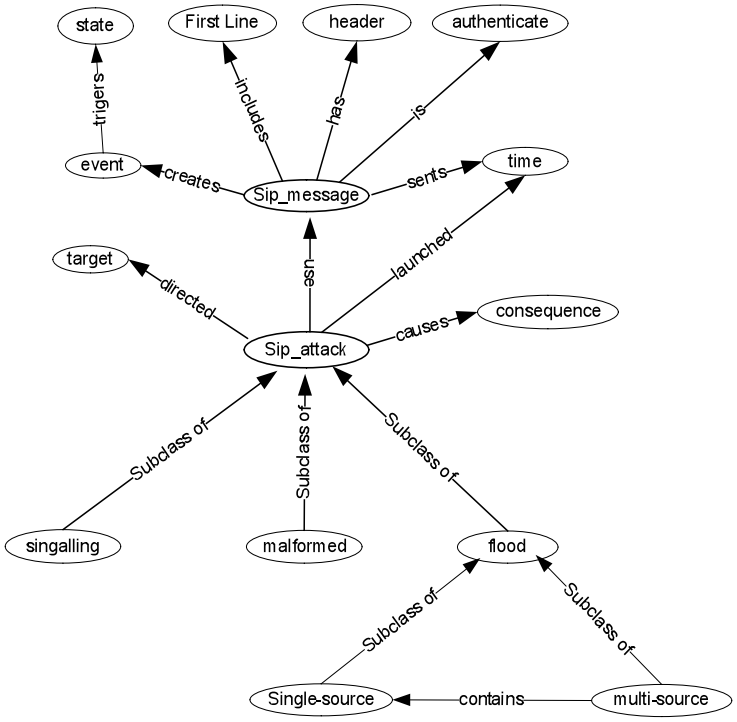

Figure 4. An Ontology based Description of Security Policies for SIP-based VoIP Services

Figure 5 illustrates a policy based on the aforementioned ontology for the SIP message REGISTER. It defines the main parts of a SIP message and the rules that should be applied to check its conformance to the specifications. More specifically, this example defines a policy of a specific SIP domain in which a SIP REGISTER message should be composed from a *REGISTER-First-Line* and from the *CSeq, From* and *To* headers, while authentication is not required since it is not explicitly described in the policy. Note, that any security requirement, which an administrator of a specific realm would like to cover, must be explicitly described in the ontology policy.

```
<sip_message id="sip_register">
        <first_line rdfresource="#REGISTER" id="1"/>
        <header rdfresource="#CSEQ"/>
        <header rdfresource="#from"/>
        <header rdfresource="#to"/>
</sip_message>
```

Figure 5. An example of the *SIP-message* sub-ontology instantiation: The REGISTER message policy

It is stressed that each of the resources described in the ontology instantiation of Figure 5 should have been defined either in a local or in a remote ontology repository. This means that the resources *First-Line: REGISTER* and *Headers: CSEQ, From, To* should be previously described, following the SIP grammar specification, as shown in Figure 6 and 7 correspondingly. Otherwise the policy would include undefined or ambiguous symbols and consequently the

ontology may not be able to interpret it appropriately. If the description of the resources already exists in a specific repository, one can reuse it by making the appropriate reference to the ontology repository.

```
<first_line ID="REGISTER">
        <method rdfresource="#REGISTER"/>
<uri>
\s+((((\d{1,3}[.]){3,3}\d{1,3}(:\d{1,5}))
(;transport[=].+)*\s+(SIP[/]\d[.]\d))|((SIP[/]\d[.]\d)\s+(\d{3})\s+.+)\s*
</uri>
</first_line>
```

Figure 6. The Definition of *First-Line Register* Resource

```
sip_message_headers rdfid="to">
        <name>#to</name>
        <rule>\s*(([''])*(\w+\s*\w*)*[''])*\s+
        ((<)*(sip:)((\w+@(\w+[.])+\w+)
        |(((\d{1,3}[.]){3,3}\d{1,3})))(>)*))(\s*;tag=.+)*\s*
        </rule>
</sip_message_headers>
<sip_message_headers rdfid="from">
        <name>
        #from
        </name>
        <rule>\s*([''])*(\w+\s*\w*)*[''])*\s+
        ((<)*(sip:)((\w+@(\w+[.])+\w+)|
        (((\d{1,3}[.]){3,3}\d{1,3})))(>)*))(\s*;tag=\w+)*(.+)*\s*
        </rule>
</sip_message_headers>
<sip_message_headers rdfid="CSEQ">
        <name>
        #CSEQ
        </name>
        <rule>
        ^\s*(CSeq:)\s*\d+\s+\b(register)\b\s*</rule>
</sip_message_headers>
```

Figure 7. The Definition of Headers: *FROM, TO, CSEQ*

2) The SIP-Attack sub-ontology

As illustrated in Figure 4, all SIP-Attacks exploit, in various ways, the SIP messages in order to cause security problems to the client provided service. The *SIP-Attack* sub-ontology (see also Part B of appendix I) comprises of the three general subclasses that follow:

- The *malformed class* corresponds to attacks exploiting messages that do not conform to the SIP message grammar in order to cause instability to the service. Such description does not require any specific formalization, as every malformed message can be seen as the complement of the corresponding well-formed message. Consequently, any message that does not comply with the SIP grammar is characterized as a malicious one. For instance, if a SIP REGISTER message does not include the header CSEQ, it is automatically characterized as malicious according to the policy presented in Figure 5.
- The *signaling class* represents attacks that illicitly terminate sessions by sending the appropriate termination message (e.g. SIP BYE, SIP CANCEL etc) on behalf of the legal user. Thus, any incoming message like SIP BYE, SIP CANCEL that cannot be authenticated should be considered as illegal and discarded according to the policy.
- The *flooding class* formalizes flooding attacks. As illustrated in Figure 4, flooding attacks are distinguished to single and multiple source flooding attacks. According to the description proposed, multiple source flooding attacks could be considered as a number of single source flooding attacks launched simultaneously. On the other hand, the identification of a single source flooding attack is based on a specific threshold. According to the RFC 3261 [7] a SIP entity can generate up to a specific number of same requests at an interval of 32 seconds. Having this threshold violated means that a flooding attack is taking place.
- The *Target class* corresponds to the available network entities that process SIP messages and are potential attack targets.
- The *Consequence class* represents the impact that a successful attack may cause to a SIP entity.

As an example, consider a case where an administrator of a SIP domain introduces a policy, similar to the one illustrated in Figure 8, for identifying REGISTER malformed messages. According to the ontology specification, for such a policy, the administrator should define the type of the attack and any additional elements (fields), which are required to fully describe it. Therefore, it is necessary to include the target of the attack (in the form of an IP address and a port number) and the possible message(s) that the attacker could exploit for launching the attack (see Appendix I & Figure 4). Specifically, in the example of Figure 8, the attack type is "Register Malformed", the target is the "registrar" and the SIP message that can be utilized is "sip_register". It is therefore clear that the inspection rules that should be employed are those defined for the "sip_register" resource (see Figures 5 to 7).

```
<Malformed id="Register-Malformed">
        <sip_message rdfresource="sip_register"/>
        <target rdfresource="registrar"/>
</Malformed>
<target id="registrar">
        <ip>195.251.145.3</ip>
        <port>5060</port>
</target>
```

Figure 8. An example of Malformed Policy

B. A Formal Representation of the SIP-Ontology in FOL

Taking into account that the basis of ontology languages like DAML+OIL [35] and OWL [36] is the FOL, we proceed in describing and formalizing the proposed ontology in *FOL*. This formalization will facilitate the integration of the proposed system with existing reasoning and inference tools [34], as well as the provision of powerful semantic ways to define robust security services for critical infrastructures like VoIP. The *FOL* transformation of the proposed ontology has been based on the guidelines presented in [37]. It should be stressed that each part of the proposed ontology corresponds to

a specific fragment of *FOL*.

1) SIP-Message FOL representation

Section V.A.1 provides the full description of the *SIP-Message* ontology. Specifically, any SIP-Message is formed by a First Line (request or response) and at least three headers. Formulas 1 to 5 correspond to the FOL fragment of a SIP message, while formulas 6 to 15 represent the relationship (i.e. authentication, sent, create, include header, has_first_line) among the SIP-message resources as illustrated in Figure 4.

$$\forall x\ SIP\_Message(x) \Leftrightarrow \exists f\ FirstLine(f) \wedge \exists^{\geq 3} h\ has\_header(h) \quad (1)$$

$$\forall f\ FirstLine(f) \Rightarrow Request(f) \vee Response(f) \quad (2)$$

$$\forall x\ Request(x) \Rightarrow \neg Response(x) \quad (3)$$

$$\forall r\ Request(r) \Leftrightarrow \exists m\ Method(m) \wedge \exists rs\ Resource(rs) \quad (4)$$

$$\forall m\ Method(m) = \{INVITE \vee REGISTER \vee OPTIONS\} \quad (5)$$

$$\forall m1,a1\ is\_auth(m1,a1) \Rightarrow authenticate(a1) \quad (6)$$

$$\forall m1,a1\ is\_auth(m1,a1) \Rightarrow SIP\_message(m1) \quad (7)$$

$$\forall m1,t1\ message\_sent(m1,t1) \Rightarrow SIP\_message(m1) \quad (8)$$

$$\forall m1,t1\ message\_sent(m1,t1) \Rightarrow time(t1) \quad (9)$$

$$\forall m1,e1\ create(m1,e1) \Rightarrow SIP\_message(m1) \quad (10)$$

$$\forall m1,e1\ create(m1,e1) \Rightarrow Event(e1) \quad (11)$$

$$\forall m,h\ has\_header(m,h) \Rightarrow SIP\_message(m) \quad (12)$$

$$\forall m,h\ has\_header(m,h) \Rightarrow header(h) \quad (13)$$

$$\forall m,f\ inc\_first\ln(m,h) \Rightarrow SIP\_message(m) \quad (14)$$

$$\forall m,f\ inc\_first\ln(m,h) \Rightarrow SIP\_message(m) \quad (15)$$

2) SIP-Attack FOL representation

As far as the *SIP-Attack* sub-ontology is concerned, each class and relationship is represented in *FOL* in a way similar to the *SIP-Message* sub-ontology. Specifically, according to the *SIP-Attack* description (see Section V.A.2: *SIP-Attack* Sub-ontology) an attack can be one of the following types: (a) malformed, (b) signaling and (c) flooding. These types of attacks, which are independent from each other, are described in *FOL* by formulas 16 to 22.

Besides, as already mentioned in Section V.A.2, a malformed message is the complement of a well-formed one (see Formula 23), while the representation of a signaling attack can take two forms: a) the existence of two or more identical SIP messages within different time frames is considered as a signaling attack (see Formula 24) and b) any message which is not authenticated, despite the fact that according to the messages's policy authentication is required, is also considered as a signaling attack (see Formula 25).

On the other hand, flooding attacks can be single or multiple source attacks (see Formula 26). A flooding attack is characterized as single source if a target receives from a specific node a number of messages that exceeds a given threshold (see Formula 27) or as multi source if the number of simultaneous single source attacks exceeds a specific threshold (see Formula 28). The remaining formulas, 29 to 39, represent the relationships among the *SIP_attack* ontology resources. Specifically, formulas 29 to 33 correspond to the subclass *property*, while formulas 33 to 39 represent the relationship among the resources *Target* and *consequence*.

$$\forall m\ SIP\_Attack(m) \Leftrightarrow Malformed(m) \vee Signalling(m) \vee Flood(m) \quad (16)$$

$$\forall m\ Malformed(m) \Rightarrow \neg Singalling(m) \quad (17)$$

$$\forall m\ Malformed(m) \Rightarrow \neg Flood(m) \quad (18)$$

$$\forall m\ Flood(m) \Rightarrow \neg Singalling(m) \quad (19)$$

$$\forall m\ Flood(m) \Rightarrow \neg Malformed(m) \quad (20)$$

$$\forall m\ Signalling(m) \Rightarrow \neg Flood(m) \quad (21)$$

$$\forall m\ Signalling(m) \Rightarrow \neg Malformed \quad (22)$$

$$\forall m\ \neg SIP\_Message(m) \Leftrightarrow Malformed(m) \quad (23)$$

$$\forall m1,m2\ SIP\_Message(m1) \wedge SIP\_Message(m2) \wedge SameAs(m1,m2) \Leftrightarrow Signalling(m1) \quad (24)$$

$$\forall m\ SIP\_Message(m) \wedge \neg Authenticate(m) \Leftrightarrow Singalling(m) \quad (25)$$

$$\forall m\ Single(m) \vee Multi(m) \Leftrightarrow Flood(m) \quad (26)$$

$$\forall m\ Single(m) \Leftrightarrow Number\_of(m) > thrshlds \wedge directed(m,t) \wedge source\_is(m,s) \quad (27)$$

$$\forall m\ Multi(m) \Leftrightarrow Number(Single(m)) > thrhsldm \quad (28)$$

$$\forall m\ Malformed(m) \Rightarrow SIP\_Attack(m) \quad (29)$$

$$\forall m\ Signalling(m) \Rightarrow SIP\_Attack(m) \quad (30)$$

$$\forall m\ Flood(m) \Rightarrow SIP\_Attack(m) \quad (31)$$

$$\forall m\ Single(m) \Rightarrow Flood(m) \quad (32)$$

$$\forall m\ Multi(m) \Rightarrow Flood(m) \quad (33)$$

$$\forall a,m\ Attack\_utilize(a,m) \Rightarrow SIP\_Attack(a) \quad (34)$$

$$\forall a,m\ Attack\_utilize(a,m) \Rightarrow SIP\_message(m) \quad (35)$$

$$\forall a,t\ attack\_target(a,t) \Rightarrow SIP\_Attack(a) \quad (36)$$

$$\forall a,t\ attack\_target(a,t) \Rightarrow target(t) \quad (37)$$

$$\forall a1,c1\ attack\_cause(a1,c1) \Rightarrow SIP\_Attack(a1) \quad (38)$$

$$\forall a1,c1\ attack\_cause(a1,c1) \Rightarrow consequence(c1) \quad (39)$$

VI. Employing the Proposed Policy In A Real Environment

The proposed ontology based policy in addition to its use for the description of security guidelines and recommendations, it can be also employed for testing the robustness of VoIP systems as well as for analyzing log files or/and raw data in order to identify the occurrence of illegal or suspicious actions. The formal representation, included in the policy, can essentially improve the effectiveness and accuracy of the identification procedure.

Consider an administrator of a SIP realm who utilizes a specific policy, like the *REGISTER Malformed* policy illustrated in Figure 8, to identify real time or offline attacks launched against the SIP registrar server. In order to facilitate the inference system to process the raw data (logs or real time traffic) it is necessary to transform them to the proposed ontology format. Let us now assume that within the raw data

there is a malformed REGISTER message like the following one:

"REGISTER AAAAA SIP/2.0"

The transformation of the above message (according to the *"SIP Message sub-ontology"* - Appendix I, Part A) produces the instance depicted in Figure 9 below. Clearly, this instantiation of the SIP REGISTER message is analogous to the REGISTER message policy illustrated in Figure 5.

```
<sip_message id="register-11">
        <first_line rdfresource="register11-fl"/>
</sip_message>
<first_line id="register11-fl>
        <method rdfresource="#REGISTER"/>
        <uri>AAAAA</uri>
</first_line>
```

Figure 9. An example of Register Instantiation

The REGISTER instantiation message is then passed to the inference tool in order to examine the compliance of the message with the SIP grammar. This is done by utilizing the defined security policy (see Figure 8) combined with the appropriate formulas that indicate the existence or not (true or false) of an attack. Specifically, the inference tool applies the regular expression defined in the 'uri' tag of the REGISTER First-Line resource (see Figure 6) to the 'uri' data of the REGISTER instantiation message, inferring that this message does not comply with the SIP grammar and thus producing a 'false' result. Using this result in conjunction with formula 23, it is concluded that the specific message is a malformed one (the negation of a false formula, turns into a true). In parallel, formula 16 evaluates to a true value, signaling the existence of an attack. A high level representation of the aforementioned attack identification procedure is provided in Figure 10.

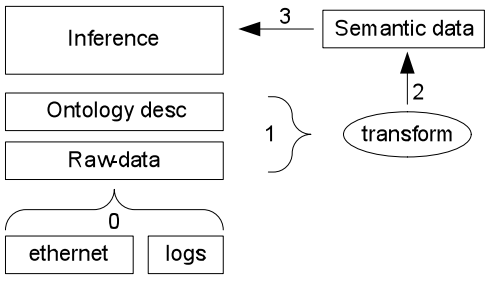

Figure 10. A general procedure to apply the ontology in Real Architecture

It is stressed that precisely the same policy and identification mechanism can be employed for the identification of an attack in real time (as demonstrated before) or off line by processing the corresponding logs.

VII. Conclusions and Future Work

The fact that VoIP architectures inherit the vulnerabilities of open networks in conjunction with the appearance of new sophisticated attacks against the signaling protocols employed for the establishment of VoIP sessions, highlights the necessity for the existence of security guidelines and best practices that can be adopted during all deployment phases of a VoIP service. In this paper such security guidelines have been proposed, represented through ontologies and, finally, transformed to a First Order Logic formal representation. It has been demonstrated that the proposed ontology-based security policy can be applied not only in a real VoIP environment for detecting attacks against a SIP service, but also for testing purposes.

The overheads introduced by the proposed policy have been found to be insignificant. However, we are currently running a set of experiments for measuring and documenting in a precise way the overall performance. Furthermore, there is on going work, investigating effective ways for the utilization of the proposed ontology-based policy in distributed VoIP environments where different providers collaborate in a many-to-many relationship model.

References

[1] Geneiatakis, D.; Dagiuklas, T.; Kambourakis, G.; Lambrinoudakis, C.; Gritzalis, S.; Ehlert, K.S.; Sisalem, D., "Survey of security vulnerabilities in session initiation protocol," Communications Surveys & Tutorials, IEEE , vol.8, no.3, pp.68-81, 3rd. Qtr. 2006

[2] Sisalem, D.; Kuthan, J.; Ehlert, S., "Denial of service attacks targeting a SIP VoIP infrastructure: attack scenarios and prevention mechanisms," Network, IEEE , vol.20, no.5, pp. 26-31, Sept.-Oct. 2006

[3] Titmus P., Securing IP telephony systems - best practices, Network SecurityVolume 2006, Issue 9, Pages 11-13, September 2006.

[4] Evern Eren, Detken Kai-Oliver, Voice over IP Security Mechanisms State of the art, risks assesment, concepts and recommendations, Internetworking 2006 ,January 2007, available on line: http://www.decoit.de/cms/upload/ pdf/IW06_eren_detken_VoIP_final.pdf

[5] Feng Cao; Malik, S., "Vulnerability analysis and best practices for adopting IP telephony in critical infrastructure sectors," Communications Magazine, IEEE , vol.44, no.4, pp. 138-145, April 2006

[6] VOIPSA, "VoIP Security and Privacy Threat Taxonomy" http://www.voipsa.org/Activities/taxonomy.php, October 2005

[7] Rosenberg, J.; Schulzrinne H.; Camarillo, G.; Johnston, A.; Peterson, J.; Spark, R.; Handley, M.; Schooler E., "Session Initiation Protocol", RFC 3261, June 2002.

[8] Geneiatakis, D.; Lambrinoudakis, C., "An ontology description for SIP security flaws", Computer Communications, Volume 30, Issue 6, , Pages 1367-1374, March 2007

[9] Darpa Internet Program Protocol Specification, "Internet Protocol", RFC 791, September 1981

[10] Darpa Internet Program Protocol Specification, "Transmission Control Protocol", RFC 793 September 1981

[11] Postel, J., User Datagram Protocol (UDP) , RFC 768, August 1980

[12] Stewart ,R.; Xie, Q.;, Morneault K.; Sharp, C.; Schwarzbauer, H.; Taylor, T.; Rytina, I.; Kalla M.; Zhang L.; Paxson V., "Stream Control Transmission Protocol", RFC 2960, October 2000,

[13] P. Mockapetris, Domain Names – Implementaion and Specification RFC 1035, November 1987

[14] Droms R.;, Dynamic Host Configuration Protocol, RFC 2131, March 1997

[15] Hersent, O.; Petit J.; Gurle D., IP Telephony: Deploying Voice-over-IP Protocols, Wiley, March 2005

[16] Andreasen, F.; Foster B.; Media Gateway Control Protocol (MGCP), RFC 3435, January 2003

[17] http://www.cisco.com/en/US/tech/tk652/tk701/tk589/tsd_technology_support_sub-protocol_home.html

[18] Schulzrinne H.; Casner S.; Frederick R.; Jacobson, V., RTP: A Transport Protocol for Real-Time Applications, July 2003

[19] Baugher, M.; McGrew, D.; Naslund, M.; Carrara E.; Norrman K., The Secure Real-time Transport Protocol (SRTP), March 2004

[20] The Zfone, http://zfoneproject.com/zrtp_ietf.html

[21] Thermos, P.; Takanen, A., Securing VoIP Networks: Threats, Vulnerabilities, and Countermeasures, Addison-Wesley Professional; August 2007
[22] Wieser, C; Laakso, M;, Schulzrinne H., Security testing of SIP implementations", available on http://compose.labri.fr/documentation/sip/Documentation/Papers/Security/Papers/462.pdf, 2003.
[23] Ruishan, Zhang.; Xinyuan, Wang.; Xiaohui, Yang.; Xuxian, Jiang.; "Billing Attacks on SIP-Based VoIP Systems" in proceedings of 1st USENIX workshop on offensive technologies, August 2007
[24] Gibson, S., "DRDoS Distributed Reflection Denial of Service,"http://grc.com/dos/drdos.htm, 2002
[25] Chen, E.Y., "Detecting DoS attacks on SIP systems", in proceedings of 1st IEEE Workshop VoIP Management and Security, Pages 53-58, April 2006
[26] Avaya., Vulnerability Issues in Implementations of the H.323 Protocol , available on line: http://support.avaya.com/elmodocs2/security/ASA-2004-002.pdf
[27] Geneiatakis D.; Lambrinoudakis C., "A Lightweight Protection Mechanism against Signaling Attacks in a SIP-Based VoIP Environment", Telecommunication Systems, Springer, to Appear
[28] Cao, F.; Jennings, C., Providing response identity and authentication in IP telephony, In proceedings of the 1st International Conference on Availability, Reliability and Security, April 2006
[29] Zhang, G.; Ehlert, S.; Magedanz T.; Sisalem D., "Denial of Service Attack and Prevention on SIP VoIP Infrastructures Using DNS Flooding", in proceeding of Principles, Systems and Applications of IP Telecommunications (IPTComm2007), July 2007
[30] Fiedler, J.; Kupka, T., Ehlert, S.; Magedanz T.; Sisalem D., "VoIP Defender: Highly Scalable SIP-based Security Architecture", Principles, Systems and Applications of IP Telecommunications (IPTComm2007), July 2007
[31] Steven T. Eckmann, Giovanni Vigna, Richard A. Kemmerer, STATL: An Attack Language for State-based Intrusion Detection, available on line: http://citeseer.ist.psu.edu/eckmann00statl.html
[32] SNORT, the de facto standard for intrusion detection/prevention. Available from: <www.snort.org>.
[33] Undercoffer, J; Joshi A.; Finin, T.; Pinkston, J. Using DAML+OIL to classify intrusive behaviours. Knowl. Eng. Rev. 18, 3, 221-241, September 2003
[34] Renamed Abox and Concept Expression Reasoner (RACER), Available on line: http://www.sts.tu-harburg.de/~r.f.moeller/racer/
[35] McGuinness D.L.; Fikes, R.; Hendler J.; Stein L.A.,; "DAML+OIL: An Ontology Language for the Semantic Web", In IEEE Intelligent Systems 17 (2002) (5), pp. 72–80
[36] McGuinness, D. L.; Frank van Harmelen, OWL Web Ontology Language, http://www.w3.org/TR/owl-features/
[37] Grosof, B. N.; Horrocks, I.; Volz, R.; Decker, S., Description logic programs: combining logic programs with description logic. In Proceedings of 12th International Conference on World Wide Web WWW '03. ACM, May 2003

APPENDIX I

Part A: SIP-Message Sub-ontology

```
<daml:Class rdf:ID=SIP_MESSAGE>
  <daml:subclassof>
    <daml:Restriction>
      <daml:onProperty rdf:resource="first_line"/>
      <daml:hasClass rdf:resource="sip_ first_line"/>
    </daml:Restriction>
  </dam:subclassof>
  <daml:subclassof>
    <daml:Restriction>
      <daml:onProperty rdf:resource="first_line"/>
      <daml:cardinality>1</daml:cardinality>
    </daml:Restriction>
  </daml:subclassof>
  <daml:subclassof>
    <daml:Restriction>
      <daml:onProperty rdf:resource="headers">
      <daml:mincardinality>3</daml:cardinality>
    </daml:Restriction>
  </daml:subclassof>
  <daml:subclassof>
    <daml:Restriction>
      <daml:onProperty rdf:resource="headers">
      <daml:range rdf:resource="sip_headers">
    </daml:Restriction>
  </daml:subclassof>
</daml>

<daml:Class rdf:ID=sip_first_line>
  <daml:disjointUnionOf parseType="daml:collection">
    <daml:Class rdf:about="request"/>
    <daml:Class rdf:about="responses"/>
  </daml:disjointUnionOf>
</daml>

<daml:DatatypeProperty rdf:ID="uri">
  <daml:domain rdf:resource="#sip_first_line"/>
  <rdf:range rdf:Resource="string"/>
</daml:DatatypeProperty>

<daml:Class rdf:ID="request">
</daml:Class>

<daml:objectProperty ref:ID="used_method">
  <daml:domain resource="#request"/>
  <daml:range rdf:resource="#methods"/>
</daml:objectProperty>

<daml:Class rdf:ID=sip_headers>
</daml>
<daml:DatatypeProperty rdf:ID="header_name">
  <daml:domain rdf:resource="#sip_headers"/>
  <rdf:range rdf:Resource="string"/>
</daml>
<daml:DatatypeProperty rdf:ID="rule">
  <daml:domain rdf:resource="#sip_ headers"/>
  <rdf:range rdf:Resource="string"/>
</daml>

<daml:Class rdf:ID="methods">
  <daml:oneOf ref:parseType="Collection">
    <daml:Thing rdf:about="REGISTER">
    <daml:Thing rdf:about="INVITE">
    <daml:Thing rdf:about="SUBSCRIBE">
    <daml:Thing rdf:about="BYE">
    <daml:Thing rdf:about="ACK">
    <daml:Thing rdf:about="CANCEL">
    <daml:Thing rdf:about="OPTIONS">
  </daml:oneof>
</daml>
```

Part B: SIP-Attack Sub-ontology

```
<daml:Class rdf:ID=attack>
```

```
</daml:Class>
<daml:ObjectProperty rdf:ID="attack_utilize">
   <daml:domain rdf:resource="#attack"/>
   <rdf:range rdf:Resource="#SIP_MESSAGE"/>
</daml:ObjectProperty>
<daml:ObjectProperty rdf:ID="attack_target">
   <daml:domain rdf:resource="#attack"/>
   <rdf:range rdf:Resource="#target"/>
</daml:ObjectProperty>

<daml:Class rdf:ID="malformed">
   <rdfs:subclassof resource="#attack"/>
   <rdfs:subclassof>
      <daml:complementof>
         <daml:Class rdf:resource=#sip_message/>
      </daml:complementof>
   </rdfs:subclassof>
</daml:Class>

 <daml:Class rdf:ID="flood">
      <rdfs:subclassof resource="attack"/>
   </daml:Class>

<daml:Class rdf:ID="single-source">
   <rdfs:subclassof resource="flood"/>
</daml:Class>

<daml:DatatypeProperty rdf:ID="source-ip">
      <daml:domain rdf:resource="#single-source">
      <rdf:range rdf:Resource="string"  />
</daml:DatatypeProperty>

<daml:DatatypeProperty rdf:ID="threshold">
      <daml:domain rdf:resource="#single-source">
      <rdf:range rdf:Resource="number"  />
</daml:DatatypeProperty>

<daml:Class rdf:ID="same-req">
      <rdfs:subclassof resource="single-source"/>
</daml:Class>

<daml:DatatypeProperty rdf:ID="session-id">
      <daml:domain rdf:resource="#same-req">
      <rdf:range rdf:Resource="string"  />
</daml:DatatypeProperty>

<daml:DatatypeProperty rdf:ID="session-to">
      <daml:domain rdf:resource="#same-req">
      <rdf:range rdf:Resource="string"  />
   </daml:DatatypeProperty>

<daml:Class rdf:ID="new-req">
      <rdfs:subclassof resource="single-source"/>
</daml:Class>

<daml:DatatypeProperty rdf:ID="session-id">
      <daml:domain rdf:resource="#new-req"/>
      <rdf:range rdf:Resource="string"  />
</daml:DatatypeProperty>

<daml:Class rdf:ID="multi-source">
      <rdfs:subclassof resource="flood"/>
</daml:Class>

<daml:ObjectProperty rdf:ID="contains">
      <daml:domain rdf:resource="multi-source"/>
      <daml:range rdf:resource=" single-source" />
</daml:ObjectProperty>

<daml:DatatypeProperty rdf:ID="memoryconsumption">
      <daml:domain rdf:resource="multi-source">
      <rdf:range rdf:Resource="number"  />
</daml:DatatypeProperty>

<daml:DatatypeProperty rdf:ID="threshold">
      <daml:domain rdf:resource="multi-source">
      <rdf:range rdf:Resource="number"  />
</daml:DatatypeProperty>

<daml:Class rdf:ID="SYN-syndrome">
      <rdfs:subclassof resource="multi-source"/>
   </daml:Class>

<daml:DatatypeProperty: ID= "without-answered">
      <daml:domain rdf:resource="SYN-syndrome"/>
      <rdf:range rdf:Resource="number"  />
</daml:DatatypeProperty>

<daml:Class rdf:ID="REF-syndrome">
      <rdfs:subclassof resource="multi-source"/>
</daml:Class>

<daml:DatatypeProperty: ID= "without-invite">
      <daml:domain rdf:resource="REF-syndrome"/>
      <rdf:range rdf:Resource="number"  />
</daml:DatatypeProperty>

<daml:Class rdf:ID="event">
</daml:Class>

<daml:ObjectProperty rdf:ID="event-uses">
      <daml:domain rdf:resource="event"/>
      <daml:range rdf:resource=" sip-message" />
</daml:ObjectProperty>

<daml:DatatypeProperty rdf:ID="event-time">
      <daml:domain rdf:resource="event"/>
      <rdf:Range rdf:Resource="string"/>
</daml:DatatypeProperty>

<daml:Class rdf:ID="state">
      <daml:oneOf ref:parseType="Collection">
         <daml:Thing rdf:about="No-state">
         <daml:Thing rdf:about="calling">
         <daml:Thing rdf:about="proceeding">
```

```
        <daml:Thing rdf:about="established">
        <daml:Thing rdf:about="terminating">
      </daml:oneof>
</daml:Class>

<daml:Class rdf:ID="singalling-attack">
      <rdfs:subclassof resource="attack"/>
      <rdfs:subclassof>
        <daml:Restriction>
          <daml:onPropertyrdf:resource="has_sip_message">
           <daml:toClass rdf:resource="SIP_Message">
        </daml:Restriction>
      </rdfs:subclassof>
      <daml:intersectionof rdf:ParseTpe="Collection">
        <daml:Class>
          <daml:complementof>
             <damlClass rdf:resource="Authenticate"/>
          </daml:complementof>
        </daml:Class>
        <daml:Restriction>
          <daml:onProperty rdf:resource="singalling-uses">
          <daml:Cardinality>2</daml:cardinality>
        </daml:Restriction>
      </daml:intersectionof>
</daml:Class>

<daml:ObjectProperty rdf:ID="singalling-uses">
      <daml:domain rdf:resource="singalling-attack"/>
      <daml:range rdf:resource=" sip-message" />
</daml:ObjectProperty>
```