HAL
archives-ouvertes.fr

Decentralized Semantic Threat Graphs

Simon Foley, William Fitzgerald

▶ To cite this version:

Simon Foley, William Fitzgerald. Decentralized Semantic Threat Graphs. 26th Conference on Data and Applications Security and Privacy (DBSec), Jul 2012, Paris, France. pp.177-192, 10.1007/978-3-642-31540-4_14 . hal-01534768

HAL Id: hal-01534768
https://hal.inria.fr/hal-01534768

Submitted on 8 Jun 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution| 4.0 International License

Decentralized Semantic Threat Graphs

Simon N. Foley[1] William M. Fitzgerald[1]

Cork Constraint Computation Centre,
Computer Science Department, University College Cork, Ireland.
s.foley@cs.ucc.ie wfitzgerald@4c.ucc.ie

Abstract. Threat knowledge-bases such as those maintained by MITRE and NIST provide a basis with which to mitigate known threats to an enterprise. These centralised knowledge-bases assume a global and uniform level of trust for all threat and countermeasure knowledge. However, in practice these knowledge-bases are composed of threats and countermeasures that originate from a number of threat providers, for example Bugtraq. As a consequence, threat knowledge consumers may only wish to trust knowledge about threats and countermeasures that have been provided by a particular provider or set of providers. In this paper, a trust management approach is taken with respect to threat knowledge-bases. This provides a basis with which to decentralize and delegate trust for knowledge about threats and their mitigation to one or more providers. Threat knowledge-bases are encoded as Semantic Threat Graphs. An ontology-based delegation scheme is proposed to manage trust across a model of distributed Semantic Threat Graph knowledge-bases.

Keywords: Decentralized Threat Management, Security Configuration

1 Introduction

Threat management is a process used to help implement a security configuration that mitigates known enterprise (security) threats. Centralised threat knowledge-bases, such as NIST's *National Vulnerability Database* (*NVD*) [1] are an integral part of the threat management process. However, in practice threat knowledge-bases are composed of threats, vulnerabilities and countermeasures that originate from multiple providers, for example US-Cert [2], Bugtraq [3] and/or vendors (such as Cisco and Microsoft). As a consequence, threat knowledge may only be trusted if it has been asserted by a particular provider or set of providers. For example, a consumer of the NVD may only want to trust knowledge about software buffer-overflow vulnerabilities that have been asserted by Bugtraq and corresponding countermeasures asserted by Microsoft. However, access to a centralised threat knowledge-base implies a global or uniform level of trust for all knowledge about threats, vulnerability and countermeasures indiscriminately.

This paper adopts a trust management approach with respect to threat knowledge-bases. The advantages are twofold. The first is that it provides a basis with which a consumer may delegate authority to trusted providers for knowledge about threats, vulnerabilities and countermeasures. Secondly, it provides

a basis to decentralize a threat knowledge-base where trusted threat knowledge may reside with the originating provider and/or be distributed across other trusted provider threat knowledge-bases.

Threat Trees/Graphs, such as [20, 28, 29], are used to represent, structure and analyse what is known about threats and their countermeasures. In this paper, threat knowledge-bases are encoded as Semantic Threat Graphs [20]. We argue that using an ontology-based framework provides a natural approach to constructing, reasoning about and managing decentralized Semantic Threat Graphs. An ontology provides a conceptual model of a domain of interest. It provides a framework for distributed and extensible structured knowledge founded on formal logic [7]. In recent years, research in computer security has seen an increase in the use of ontologies. For example, ontologies have been applied to the areas of information security (common security vocabulary) [24], security management (threats, vulnerabilities and countermeasures) [17], access control [15], policy management [25] and trust management [33]. The decentralized Semantic Threat Graphs (ontology fragments) are implemented in OWL-DL, a language subset of OWL which is a W3C standard that includes Description Logic reasoning semantics [30].

Distributed fragments of Semantic Threat Graphs that are naturally composable under the open-world assumption, provide a unified view of the threat-based domain. Ontologies provide for separation of concerns, whereby consumers and providers of threat-based knowledge can be separately developed, with reasoning and deployment over their composition done locally. It also means that information about new threats, vulnerabilities and countermeasures can be incorporated as new facts within existing Semantic Threat Graph knowledge-bases.

The paper is outlined as follows. Section 2 provides an overview of Description Logic. Section 3 outlines the Semantic Threat Graphs model and an encoding of standards of best practice for threat mitigation. Section 4 describes the delegation scheme used to manage trust across a model of distributed threat knowledge-bases. A model for decentralized Semantic Threat Graphs is presented in Section 5. Section 6 provides use case scenarios that demonstrate how the approach may work in practice. Related research is discussed in Section 7.

2 Description Logic and Knowledge-Bases

The Description Logic $\mathcal{SHOIN}^{(\mathcal{D})}$ is a decidable portion of First Order Logic that can be used to represent and reason about application knowledge and is commonly implemented as OWL-DL [7]. Knowledge is described in terms of *concepts* about *individuals* and their relationships. For example, suppose that concept `DOS` describes denial of service threats and concept `TCPcntr` describes TCP-stack based countermeasures such as `syn-cookie` and `syn-cache`, then the concept

$$(\exists_1 \mathit{mitigates}.\texttt{DOS}) \sqcap \texttt{TCPcntr}$$

can be considered to characterize `TCPcntr` countermeasure concepts that mitigate denial of service threats. In this case *mitigates* is a *property* that has been

defined between DOS threats and countermeasures. For example, the individual countermeasure syn-cache is related (property *mitigates*) to the individual DOS threat syn-flood. A Description Logic concept corresponds to a unary predicate; intuitively, it represents a set, and concept conjunction ($\sqcap$) and disjunction ($\sqcup$) provide set intersection and union, respectively. A Description Logic property (role) corresponds to a binary predicate. The concept ($\exists_{\geq 1}\ p.\phi$) defines individuals related to at least one individual of concept ϕ via property p.

A knowledge-base comprises a terminological component, hereafter referred to as its *TBox*, and an assertional component (its *ABox*). In addition to describing concepts, a TBox may define concept and property relationships. These are given as axioms of the form $\phi_1 \sqsubseteq \phi_2$, given concepts ϕ_1 and ϕ_2 and where subsumption $\sqsubseteq$ can be interpreted as subset. For example, the TBox containing the concepts from the previous paragraph could include the axiom $\texttt{DOS} \sqsubseteq \texttt{Threat}$, indicating that denial of service is a kind of threat. Property subsumption axioms may similarly be defined, which we also represent as $\phi_1 \sqsubseteq \phi_2$ if no ambiguity arises.

3 Semantic Threat Graphs

Semantic Threat Graphs [20], a variation of the traditional Threat/Attack Tree, are encoded within an ontology-based framework. Figure 1 provides an abstract model for Semantic Threat Graphs and identifies the key concepts and relationships.

Enterprise IT assets are represented as individuals of the *Asset* concept. An asset may have one or more *hasWeakness*'s (property relationship) that relate to individuals categorised in the *Vulnerability* concept. Individuals of the *Vulnerability* concept are exploitable (*exploitedBy*) by a threat or set of threats (*Threat* concept). As a consequence, an asset that has a vulnerability is, therefore, also *threatenedBy* a corresponding *Threat*. A countermeasure *mitigates* particular vulnerabilities. Countermeasures are deemed to be kinds-of assets and thus are defined as a *subConceptOf Asset*.

Semantic Threat Graphs can be used to encode standards and best practices for threat mitigation using firewalls [20]. Mis-configured firewall security configurations have the potential to expose both internal servers and the firewalls themselves to threats. For example, consider the following scenario where a webServer is susceptible to a $\texttt{threat}_{\texttt{SynFlood}}$ attack via the $\texttt{vul}_{\texttt{WebTCPConnOverflow}}$ weakness. An individual $\texttt{vul}_{\texttt{WebTCPConnOverflow}}$ is representative of a weakness in the TCP stack where it is possible exceed the maximum number of socket connections permitted by the TCP protocol due to a syn flood attack [32]. An iptables rule, represented as individual $\texttt{iptr}_{\texttt{SynThresholdWeb}}$, mitigates the vulnerability $\texttt{vul}_{\texttt{WebTCPConnOverflow}}$ on the Web server (webServer). Note that we use human-interpretable names (in a typewriter font) for individuals in the ontology as a way of suggesting their meaning. For example, individual $\texttt{iptr}_{\texttt{SynThresholdWeb}}$ represents an iptables rule (iptr) that limits TCP syn packet connections to the Web server (SynThresholdWeb).

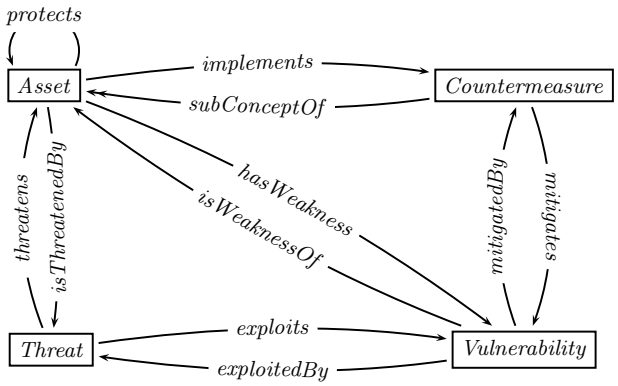

Fig. 1. Abstract Semantic Threat Graph Model.

The Semantic Threat Graph model presented in Figure 1 can be further refined with sub-concepts. For example, the *Threat* concept can define a number of sub-concepts in accordance with best practice, such as the Microsoft STRIDE standard whereby threats are categorised as (Figure 2): ***S**poofing identity*, ***T**ampering with data*, ***R**epudiation*, ***I**nformation disclosure*, ***D**enial of service* and ***E**levation of privilege* [23]. A similar hierarchy is adopted for the corresponding vulnerability and countermeasure concepts.

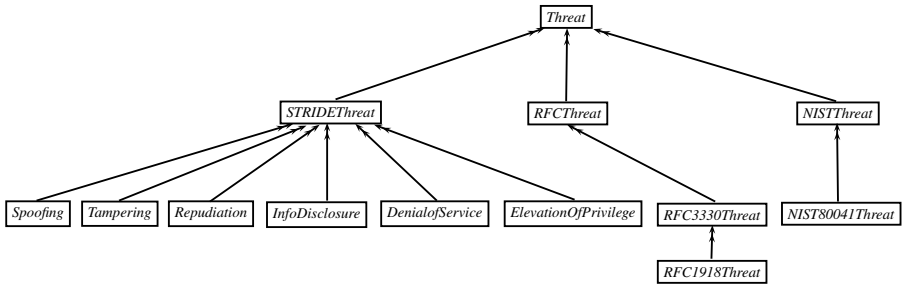

Fig. 2. Fragment of Threat Hierarchy.

A best practice standard is a high-level document, written in natural language (typically English text), that defines a set of recommended best practices (countermeasures) to protect sensitive and critical system resources. Best practice standards for network access control, including NIST for secure Web-servers [34] and Internet RFCs for anti-bogon (RFC3330) are encoded as Semantic Threat Graphs (ontologies). How these best practice standards are encoded in terms of Semantic Threat Graphs is described in [20]. For example, Table 1 provides a Semantic Threat Graph interpretation for part of the NIST-800-41 standard [35] for

firewall configuration. FBP-1 recommends that (spoofed) packets arriving on an external interface claiming to have originated from either of the three RFC1918 reserved internal IP address ranges should be dropped. Such traffic indicates a denial of service attack typically involving the TCP syn flag. Therefore, *Threat* individual $\texttt{threat}_{\texttt{Inbound192.168.0.0/16SrcIPPkt}}$, is asserted to be a member of subconcepts *Spoofing*, *DenialOfService* and $RFC1918Threat$. Figure 2 illustrates a partial hierarchy of threats.

ID	**Recommendation Description**		
FBP-1	Deny "*Inbound or Outbound traffic from a system using a source address that falls within the address ranges set aside in RFC1918 as being reserved for private networks*" [35].		
	Threat	**Vulnerability**	**Countermeasure**
	$\texttt{threat}_{\texttt{Inbound192.168.0.0/16SrcIPPkt}}$	$\texttt{vul}_{\texttt{UnAuthenInbound192.168.0.0/16PktToFW}}$	$\texttt{iptr}_{\texttt{DropIn192.168.0.0/16SrcIPPktInputChain}}$
	$\texttt{threat}_{\texttt{Outbound192.168.0.0/16SrcIPPkt}}$	$\texttt{vul}_{\texttt{UnAuthenOutbound192.168.0.0/16PktFromFW}}$	$\texttt{iptr}_{\texttt{DropOut192.168.0.0/16SrcIPPktOutputChain}}$
	$\texttt{threat}_{\texttt{Inbound192.168.0.0/16SrcIPPkt}}$	$\texttt{vul}_{\texttt{UnAuthenInbound192.168.0.0/16PktToHost}}$	$\texttt{iptr}_{\texttt{DropIn192.168.0.0/16SrcIPPktForwardChain}}$
	$\texttt{threat}_{\texttt{Outbound192.168.0.0/16SrcIPPkt}}$	$\texttt{vul}_{\texttt{UnAuthenOutbound192.168.0.0/16PktFromHost}}$	$\texttt{iptr}_{\texttt{DropOut192.168.0.0/16SrcIPPktForwardChain}}$
	$\texttt{threat}_{\texttt{Inbound10.0.0.0/8SrcIPPkt}}$	$\texttt{vul}_{\texttt{UnAuthenInbound10.0.0.0/8PktToFW}}$	$\texttt{iptr}_{\texttt{DropIn10.0.0.0/8SrcIPPktInputChain}}$
	$\texttt{threat}_{\texttt{Outbound10.0.0.0/8SrcIPPkt}}$	$\texttt{vul}_{\texttt{UnAuthenOutbound10.0.0.0/8PktFromFW}}$	$\texttt{iptr}_{\texttt{DropOut10.0.0.0/8SrcIPPktOutputChain}}$
	$\texttt{threat}_{\texttt{Inbound10.0.0.0/8SrcIPPkt}}$	$\texttt{vul}_{\texttt{UnAuthenInbound10.0.0.0/8PktToHost}}$	$\texttt{iptr}_{\texttt{DropIn10.0.0.0/8SrcIPPktForwardChain}}$
	$\texttt{threat}_{\texttt{Outbound10.0.0.0/8SrcIPPkt}}$	$\texttt{vul}_{\texttt{UnAuthenOutbound10.0.0.0/8PktFromHost}}$	$\texttt{iptr}_{\texttt{DropOut10.0.0.0/8SrcIPPktForwardChain}}$
	$\texttt{threat}_{\texttt{Inbound172.16.0.0/12SrcIPPkt}}$	$\texttt{vul}_{\texttt{UnAuthenInbound172.16.0.0/12PktToFW}}$	$\texttt{iptr}_{\texttt{DropIn172.16.0.0/12SrcIPPktInputChain}}$
	$\texttt{threat}_{\texttt{Outbound172.16.0.0/12SrcIPPkt}}$	$\texttt{vul}_{\texttt{UnAuthenOutbound172.16.0.0/12PktFromFW}}$	$\texttt{iptr}_{\texttt{DropOut172.16.0.0/12SrcIPPktOutputChain}}$
	$\texttt{threat}_{\texttt{Inbound172.16.0.0/12SrcIPPkt}}$	$\texttt{vul}_{\texttt{UnAuthenInbound172.16.0.0/12PktToHost}}$	$\texttt{iptr}_{\texttt{DropIn172.16.0.0/12SrcIPPktForwardChain}}$
	$\texttt{threat}_{\texttt{Outbound172.16.0.0/12SrcIPPkt}}$	$\texttt{vul}_{\texttt{UnAuthenOutbound172.16.0.0/12PktFromHost}}$	$\texttt{iptr}_{\texttt{DropOut172.16.0.0/12SrcIPPktForwardChain}}$
ID	**Recommendation Description**		
FBP-2	Deny "*Inbound traffic containing ICMP (Internet Control Message Protocol) traffic*" [35].		
	Threat	**Vulnerability**	**Countermeasure**
	$\texttt{threat}_{\texttt{ICMPNetworkScan}}$	$\texttt{vul}_{\texttt{InfoDisclosureICMPReplyPktFromFW}}$	$\texttt{iptr}_{\texttt{DropInICMPPktInputChain}}$
	$\texttt{threat}_{\texttt{ICMPNetworkScan}}$	$\texttt{vul}_{\texttt{InfoDisclosureICMPReplyPktFromHost}}$	$\texttt{iptr}_{\texttt{DropInICMPPktForwardChain}}$

Table 1. Ontology Extract for NIST-800-41: Guidelines on Firewalls & Firewall Policy.

4 Knowledge Delegation as Subsumption

A (distributed) system may comprise of a number of separately managed knowledge bases. Each knowledge-base is assumed to have a unique name that indicates its controlling/owning principal. We assume that each atomic concept (or property) ϕ syntactically embeds the name $(\phi)^{\mathcal{N}}$ of the principal that describes the concept (or property). For example, a TBox owned by principal A includes a concept $\texttt{A:DOS}$ where $(\texttt{A:DOS})^{\mathcal{N}} = A$. A principal P has *jurisdiction* over any concept (or property) ϕ if $(\phi)^{\mathcal{N}} = P$; this means that P is considered to be the original authority on ϕ.

Note that while a TBox may contain concepts originating from different principals, all concepts referenced in a concept expression are required to have the same name. For example, $(\texttt{A:Threat} \sqcap \texttt{A:DOS})^{\mathcal{N}} = A$. This ensures a consistent interpretation for our *syntatic* approach to referencing (the originator of) concepts. Future research will consider how a permission-naming logic such as [21] can be used to provide a consistent treatment for the originators of a concept such as $(\texttt{A:Threat} \sqcap \texttt{B:DOS})$.

Principals may make public assertions about terminological knowledge. A public assertion $P \mid\!\approx (\phi_1 \sqsubseteq \phi_2)$ is a statement by principal P about concept (or property) subsumption. For example, $A \mid\!\approx (\texttt{B:DOS} \sqsubseteq \texttt{A:Threat})$ is an assertion by A that its concept of threat includes B's concept of denial of service. These ontology mappings can be used to implement delegation of jurisdiction. Given $(\phi_1)^{\mathcal{N}} = Q$ and $(\phi_2)^{\mathcal{N}} = P$ then $P \mid\!\approx (\phi_1 \sqsubseteq \phi_2)$ is an assertion by P that it trusts Q when it describes ϕ_1 to the extent that Q's concept of ϕ_1 can be considered as a kind of ϕ_2 concept over which P's has jurisdiction. For example, suppose that principal B provides a vulnerability reporting service, then assertions $A \mid\!\approx (\texttt{B:DOS} \sqsubseteq \texttt{A:DOS})$ and $A \mid\!\approx (\texttt{B:mitigates} \sqsubseteq \texttt{A:mitigates})$ mean that A trusts B's mitigation knowledge for denial of service attacks.

Transitive subsumption in $\mathcal{SHOIN}^{(\mathcal{D})}$ can be used to reason over chains of delegation statements. For example, public assertion $B \mid\!\approx (\texttt{C:mitigates} \sqsubseteq \texttt{B:mitigates})$ indicates that the vulnerability reporting service B trusts mitigation recommendations provided by a software developer C. Continuing the example, principal A can use these public assertions to deduce that $\texttt{C:mitigates} \sqsubseteq \texttt{A:mitigates}$ and thus be happy to trust mitigation recommendations from the software developer.

The following rule defines the conditions under which an arbitrary principal may import, into its TBox, a public assertion (delegation) from P.

$$\frac{P \mid\!\approx (\phi_1 \sqsubseteq \phi_2); (\phi_2)^{\mathcal{N}} = P}{\texttt{import } \phi_1 \sqsubseteq \phi_2 \texttt{ into TBox}}$$

This does not extend the semantics of $\mathcal{SHOIN}^{(\mathcal{D})}$, rather, it is a syntactic treatment whereby delegation statements translate to concept axioms that can in turn be reasoned over within $\mathcal{SHOIN}^{(\mathcal{D})}$. This treatment is easily modeled within OWL-DL. The URI of an OWL-DL document provides its principal/namespace. A public assertion $P \mid\!\approx (\phi_1 \sqsubseteq \phi_2)$ is an ontology document that is trusted to originate from the namespace of P: this trust can be achieved by P signing the document. The ontology-import constructor `owl:imports` is used to import a public assertion that is confirmed to come from a from another namespace.

The assertional component of a knowledge base, hereafter referred to as its *ABox*, contains assertions about named concept individuals. A concept assertion $\phi(i)$, indicates that named individual i is a member of concept ϕ; a role (property) assertion $p(i, j)$ indicates that named individual i is related to named individual j under property p. For example, ABox assertion `DOS(syn-flood)` describes that individual `syn-flood` is a `DOS` threat and ABox role assertion `mitigates(syn-cache,syn-flood)` describes that the `syn-cache` countermeasure mitigates a `syn-flood` threat.

We use a similar naming scheme for individuals whereby $(i)^{\mathcal{N}}$ indicates a principal/namespace syntactically embedded in the identifier of the individual i. Principals may make public assertions about individuals. A public assertion $P \mid\!\sim \phi(i)$ is a statement by P that named individual i is a member of concept ϕ. A principal may not make public assertions about ABox knowledge (concept and individual) that is not under its jurisdiction. However, a principal may use

subsumption to infer ABox knowledge that is effectively under the jurisdiction of others. The following rule defines the conditions under which an arbitrary principal may import, into its ABox, a public assertion from P about a named individual i and concept ϕ.

$$\frac{P \mid\sim \phi(i); (\phi)^{\mathcal{N}} = (i)^{\mathcal{N}} = P}{\texttt{import } \phi(i) \texttt{ into ABox}}$$

A similar rule can be defined for public ABox property assertions.

Continuing the above example, $\texttt{B} \mid\sim(\texttt{B:DOS}(\texttt{B:syn-flood}))$ is a statement by the vulnerability reporting service `B` that `B:syn-flood` is a `B:DOS` threat. On importing this ABox assertion and the TBox assertion $\texttt{A} \mid\approx(\texttt{B:DOS} \sqsubseteq \texttt{A:DOS})$, it is possible for `A` to deduce $\texttt{A:DOS}(\texttt{B:syn-flood})$, that is, `B:syn-flood` is an `A:DOS` threat.

These public ABox assertions are a syntactic treatment that do not extend $\mathcal{SHOIN}^{(\mathcal{D})}$. In practice, OWL-DL individuals include reference to their namespace (principal) and a public ABox assertion $P \mid\sim \phi(i)$ is an ontology document that has been signed by its originator P and a valid $\phi(i)$ can be imported into another ontology using the `owl:imports` constructor.

In general, a public TBox assertion $P \mid\approx(\phi_1 \sqsubseteq \phi_2)$ is effectively a delegation certificate that can be understood as a statement $P \mid\approx(\phi_2 \sqsupset \phi_1)$ in a delegation logic such as [4], where ϕ_1 and ϕ_2 are unary (or binary) predicates that refer to static principals $(\phi_1)^{\mathcal{N}}$ and $(\phi_2)^{\mathcal{N}}$, respectively. A public ABox assertion $P \mid\sim \phi(i)$ is a signed value of type ϕ that can be effectively considered to be a form of a signed permission that cannot be forged by another principal that does not hold jurisdiction.

Note that for ease of presentation, delegation of trust is assumed transitive. Non-transitive trust can be incorporated into the model, for example, by adding a SPKI-style [16] delegation-bit to delegation certificates. Alternatively, a KeyNote-style [12] Action_Authorizers concept could be added to the ontology to constrain the delegation.

5 Delegation in Semantic Threat Graphs

A decentralized Semantic Threat Graph (STG) uses subsumption to model the delegation of jurisdiction over (potentially distributed) fragments of a Semantic Threat Graph. These fragments can include concepts such as $Vulnerability$, $Threat$ and $Countermeasure$, assertions about membership of these concepts and assertions about properties between the concepts. In this section, we outline how the delegation involving these fragments is encoded using subsumption; Section 6 will provide complete examples of decentralized Semantic Threat Graphs.

STG Concept Delegation. Subsumption is used to define delegation of $Threat$, $Vulnerability$ and $Countermeasure$ concepts between principals. For example,

$$Y \mid\approx X : DenialOfService \sqsubseteq Y : Threat$$

defines that principal Y trusts principal X, concerning the identification (naming) of denial of service attacks. Suppose that X in turn asserts

$$X \mid\sim X : \mathit{DenialOfService}(X : \mathtt{threat}_{\mathtt{SynFlood}})$$

that is, $X : \mathtt{threat}_{\mathtt{SynFlood}}$ is a *DenialOfService* individual, as identified by X. In this case, as a result of the delegation by subsumption above, principal Y can safely deduce that

$$Y : \mathit{Threat}(X : \mathtt{threat}_{\mathtt{SynFlood}})$$

that is, $X : \mathtt{threat}_{\mathtt{SynFlood}}$ can be identified as an individual of Y's *Threat* concept. Similar assertions can be made about other Semantic Threat concepts including $Vulnerability$ and $Countermeasure$.

STG Property Delegation. Like concepts, properties can be hierarchical, and over which, principals may make jurisdiction assertions. For example, the property delegation

$$Y \mid\approx X : \mathit{exploits} \sqsubseteq Y : \mathit{exploits}$$

is a statement by Y that it is willing to trust properties from the knowledge-base of X that relate how vulnerabilities are exploited by threats. For example, suppose that principal X asserts

$$X \mid\sim X : \mathit{exploits}(X : \mathtt{threat}_{\mathtt{SynFlood}}, X : \mathtt{vul}_{\mathtt{WebTCPConnMax}})$$

then principal Y, trusting X's assertions on exploits ($X : \mathit{exploits} \sqsubseteq Y : \mathit{exploits}$) can infer the following statements in its knowledge base.

$$Y : \mathit{Threat}(X : \mathtt{threat}_{\mathtt{SynFlood}}), Y : \mathit{Vulnerability}(X : \mathtt{vul}_{\mathtt{WebTCPConnMax}}),$$
$$Y : \mathit{exploits}(X : \mathtt{threat}_{\mathtt{SynFlood}}, X : \mathtt{vul}_{\mathtt{WebTCPConnMax}})$$

STG Property Restriction Delegation. Restrictions ('quantifier' and 'hasValue') can be applied to properties and are used to constrain an individual's membership to a specific concept. A property restriction effectively describes an anonymous or unnamed concept that contains all the individuals that satisfy the restriction. For example, principal Y asserts

$$Y \mid\approx (\exists_{\geq 1} X : \mathit{exploits}.X : Vulnerability \sqsubseteq Y : Threat)$$

meaning that threat individuals that are members of an unnamed threat concept $\exists_{\geq 1}\ X : \mathit{exploits}.X : Vulnerability$ defined within principal X's knowledge-base are considered as trusted individuals of concept $Y : Threat$ in principal Y's knowledge-base.

A 'hasValue' restriction, denoted by $\ni$, describes a set of individuals that are members of an anonymous concept (domain) that are related to a specific individual along the range of a given property. For example, in

$$Y \mid\approx X : \mathit{exploits} \ni X : \mathtt{vul}_{\mathtt{WebTCPConnMax}} \sqsubseteq Y : Threat$$

Y asserts that it trusts principal X's knowledge about threats that exploit the $X : \texttt{vul}_{\texttt{WebTCPConnMax}}$. Therefore, threat is $X : \texttt{threat}_{\texttt{SynFlood}}$ is considered a threat (a member of concept $Y : Threat$) within principal Y's knowledge-base. Note, concept $X : exploits \ni X : \texttt{vul}_{\texttt{WebTCPConnMax}}$ is a sub-concept of the $\exists_{\geq 1}\ X : exploits.X : Vulnerability$.

6 Decentralized Semantic Threat Graphs: Use Cases

In this section, a series of examples are presented in order to demonstrate how decentralized Semantic Threat Graphs may work in practice and in which we identify some potential business usage patterns (Figure 3). We believe this to be an improvement over the current centralised approach where trust between different providers is not discriminated.

Knowledge about threats, vulnerabilities and countermeasures while originating from a number of providers, is typically managed within centralised threat knowledge-bases such as NIST's *National Vulnerability Database* (*NVD*) [1]. The advantage of decentralising a threat knowledge-base means that threat knowledge may reside with the originating provider and/or be distributed across other trusted third party providers. A disadvantage of a centralised approach is that it implies on the part of a consumer a global level of trust for all threat knowledge indiscriminately. However, a consumer may only wish to trust threat knowledge that has been provided by a particular provider or set of providers. For example, a consumer of the NVD may only want to trust knowledge about network-based Denial of Service threats that have been provided by US-Cert and corresponding (firewall) countermeasures provided by Redhat. The advantage of a trust management approach is that it provides a basis with which a consumer may delegate authority to trusted providers for threat knowledge. In practice, providers/producers construct fragments of Semantic Threat Graphs and consumers make assertions about the conditions under which they trust the fragments provided by providers. These are defined as a combination of STG Concept, Property and Property Restriction delegation assertions described in the previous section.

Use Case 1. A company (for example, *ACME* Inc.) having a consumer role, delegates jurisdiction for knowledge about threats, vulnerabilities and countermeasures to one or more trusted compliance agencies (for example, *NIST*) having a role of a provider. A compliance agency, having a consumer role in this instance, in turn delegates jurisdiction for knowledge about threats, vulnerabilities and countermeasures to one or more trusted vendors (for example Redhat). A vendor (provider) may then notify the company with respect to Semantic Threat Graph knowledge it has jurisdiction over. The company can import these STG assertions into its knowledge-base and use the knowledge during reasoning. For

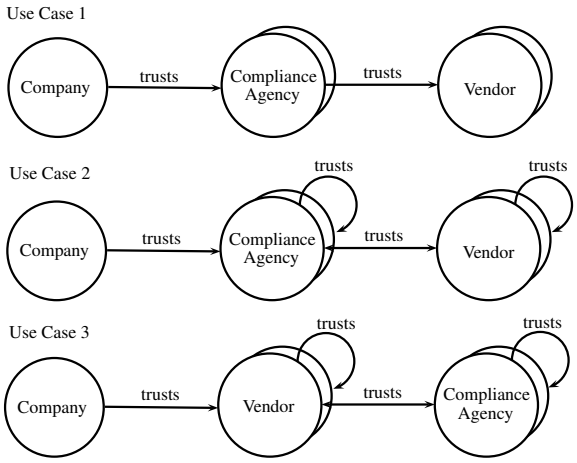

Fig. 3. Example Delegation Usage Patterns.

example, principal *ACME* asserts:

$$\begin{aligned} NIST\text{:}\, NIST80041Threat &\sqsubseteq ACME\text{:}\, Threat \\ NIST\text{:}\, NIST80041Vulnerability &\sqsubseteq ACME\text{:}\, Vulnerability \\ NIST\text{:}\, NIST80041Countermeasure &\sqsubseteq ACME\text{:}\, Countermeasure \\ NIST\text{:}\, exploits &\sqsubseteq ACME\text{:}\, exploits \\ NIST\text{:}\, mitigates &\sqsubseteq ACME\text{:}\, mitigates \end{aligned}$$

indicating that it trusts countermeasure recommendations made by *NIST* regarding threats and vulnerabilities that are to be mitigated in order to be be compliant with NIST-800-41 [35] firewall best practice. Note that, when no ambiguity can arise, we drop the turnstile notation "$\approx$" and "$\sim$" and infer the principal from the statement.

Principal *NIST*, in turn, asserts:

$$\begin{aligned} NIST\text{:}\, Spoofing &\sqsubseteq NIST\text{:}\, NIST80041Threat \\ NIST\text{:}\, UnAuthPkt &\sqsubseteq NIST\text{:}\, NIST80041Vulnerability \\ RH\text{:}\, Spoofing &\sqsubseteq NIST\text{:}\, Spoofing \\ RH\text{:}\, UnAuthPkt &\sqsubseteq NIST\text{:}\, UnAuthPkt \\ RH\text{:}\, IptablesRule &\sqsubseteq NIST\text{:}\, NIST80041Countermeasure \\ RH\text{:}\, exploits &\sqsubseteq NIST\text{:}\, exploits \\ RH\text{:}\, mitigates &\sqsubseteq NIST\text{:}\, mitigates \end{aligned}$$

indicating that principal *RH* (Redhat vendor) is trusted to specify Linux iptables firewall countermeasures used to mitigate spoofing threats and associated vulnerabilities. Intuitively, the NIST compliance agency has outsourced the instantiation of spoof-based threats and recommended firewall countermeasures to

Redhat. Note, the delegation statement made by *NIST* also includes additional knowledge about its threat and vulnerability hierarchy. For example concept $NIST{:}\,Spoofing$ is a subsumed by concept $NIST{:}\,NIST80041Threat$.

Delegation chains (transitive subsumption) constructed in terms of concept and property subsumption can be reasoned over within OWL-DL to establish if received ABox statements are to be trusted and imported. For example, Redhat (principal RH) asserts the following anti-bogon IP spoofing threat information:

$$
\begin{aligned}
&RH{:}\,Spoofing(RH{:}\,\texttt{threat}_{\texttt{Inbound192.168.0.0/16SrcIPPkt}}),\\
&RH{:}\,UnAuthPkt(RH{:}\,\texttt{vul}_{\texttt{UnAuthenInbound192.168.0.0/16PktToHost}}),\\
&RH{:}\,IptablesRule(RH{:}\,\texttt{iptr}_{\texttt{DropIn192.168.0.0/16SrcIPPktForwardChain}}),\\
&RH{:}\,exploits(RH{:}\,\texttt{threat}_{\texttt{Inbound192.168.0.0/16SrcIPPkt}},\\
&\qquad\qquad RH{:}\,\texttt{vul}_{\texttt{UnAuthenInbound192.168.0.0/16PktToHost}}),\\
&RH{:}\,mitigates(RH:\,\texttt{iptr}_{\texttt{DropIn192.168.0.0/16SrcIPPktForwardChain}},\\
&\qquad\qquad RH:\,\texttt{vul}_{\texttt{UnAuthenInbound192.168.0.0/16PktToHost}})
\end{aligned}
$$

On receipt of this ABox statement principal $ACME$ has no prior trust relationship with RH. However, given the set of known delegation statements, $ACME$ can form the following delegation (trust) chains.

$$
\begin{aligned}
&RH{:}\,Spoofing \sqsubseteq NIST{:}\,Spoofing \sqsubseteq NIST{:}\,NIST80041Threat \sqsubseteq ACME{:}\,Threat\\
&RH{:}\,UnAuthPkt \sqsubseteq NIST{:}\,UnAuthPkt \sqsubseteq NIST{:}\,NIST80041Vul \sqsubseteq ACME{:}\,Vulnerability\\
&RH{:}\,IptablesRule \sqsubseteq NIST{:}\,NIST80041Countermeasure \sqsubseteq ACME{:}\,Countermeasure\\
&RH{:}\,exploits \sqsubseteq NIST{:}\,exploits \sqsubseteq ACME{:}\,exploits\\
&RH{:}\,mitigates \sqsubseteq NIST{:}\,mitigates \sqsubseteq ACME{:}\,exploits
\end{aligned}
$$

As a consequence, $ACME$ can deduce that the ABox mitigation knowledge received from RH is trusted. It then becomes possible for $ACME$ to deduce a new concept hierarchy within its local Semantic Threat Graph knowledge-base, for example:

$$RH{:}\,Spoofing \sqsubseteq NIST{:}\,Spoofing \sqsubseteq NIST{:}\,NIST80041Threat \sqsubseteq ACME{:}\,Threat$$

in addition to the following inferred concept membership and property relations:

$$
\begin{aligned}
&ACME{:}\,Threat(RH{:}\,\texttt{threat}_{\texttt{Inbound192.168.0.0/16SrcIPPkt}}),\\
&ACME{:}\,Vulnerability(RH{:}\,\texttt{vul}_{\texttt{UnAuthenInbound192.168.0.0/16PktToHost}}),\\
&ACME{:}\,Countermeasure(RH{:}\,\texttt{iptr}_{\texttt{DropIn192.168.0.0/16SrcIPPktForwardChain}}),\\
&ACME{:}\,exploits(RH{:}\,\texttt{threat}_{\texttt{Inbound192.168.0.0/16SrcIPPkt}},\\
&\qquad\qquad RH{:}\,\texttt{vul}_{\texttt{UnAuthenInbound192.168.0.0/16PktToHost}}),\\
&ACME{:}\,mitigates(RH{:}\,\texttt{iptr}_{\texttt{DropIn192.168.0.0/16SrcIPPktForwardChain}},\\
&\qquad\qquad RH{:}\,\texttt{vul}_{\texttt{UnAuthenInbound192.168.0.0/16PktToHost}})
\end{aligned}
$$

Use Case 2. As in the previous use case, a company may delegate jurisdiction for knowledge about threats, vulnerabilities and countermeasures to one or more trusted compliance agencies. However, rather than a compliance agency delegating jurisdiction over threats, vulnerabilities and countermeasures as a collection

to one or more vendors such as Redhat or Cisco, it may instead decide to delegate certain fragments (for example threats) to one or more additional compliance agencies and other fragments (for example countermeasures) to one or more vendors. Vendors in turn may also trust one or more compliance agencies.

In this example, *ACME* makes the same TBox statement for delegation of jurisdiction to *NIST* defined in the previous scenario. Principal *NIST* delegates jurisdiction to *CVE* (compliance agency) for knowledge about NIST-800-41 spoofing threats and vulnerabilities only.

$$\begin{aligned} NIST\text{:}\, Spoofing &\sqsubseteq NIST\text{:}\, NIST80041Threat \\ NIST\text{:}\, UnAuthPkt &\sqsubseteq NIST\text{:}\, NIST80041Vulnerability \\ CVE\text{:}\, Spoofing &\sqsubseteq NIST\text{:}\, Spoofing \\ CVE\text{:}\, UnAuthPkt &\sqsubseteq NIST\text{:}\, UnAuthPkt \\ CVE\text{:}\, exploits &\sqsubseteq NIST\text{:}\, exploits \end{aligned}$$

Principal *NIST* also asserts the following delegation statement stating that principal *RH* is trusted for NIST-800-41 based iptables firewall countermeasures.

$$\begin{aligned} RH\text{:}\, iptablesRule &\sqsubseteq NIST\text{:}\, NIST80041Countermeasure \\ RH\text{:}\, mitigates &\sqsubseteq NIST\text{:}\, mitigates \end{aligned}$$

Note, trust is not bidirectional. Given that *RH* has not been given jurisdiction over relevant threats and vulnerabilities with which to make iptables recommendations, it must also delegate jurisdiction to *NIST* for this knowledge.

$$\begin{aligned} NIST\text{:}\, Spoofing &\sqsubseteq RH\text{:}\, Spoofing \\ NIST\text{:}\, UnAuthPkt &\sqsubseteq RH\text{:}\, UnAuthPkt \\ NIST\text{:}\, exploits &\sqsubseteq RH\text{:}\, exploits \end{aligned}$$

Principal *RH* receives the following ABox statements from *CVE* for which it has no prior trust relationship.

$$\begin{aligned} &CVE\text{:}\, Spoofing(CVE\text{:}\, \mathtt{threat}_{\mathtt{Inbound192.168.0.0/16SrcIPPkt}}), \\ &CVE\text{:}\, UnAuthPkt(CVE\text{:}\, \mathtt{vul}_{\mathtt{UnAuthenInbound192.168.0.0/16PktToHost}}), \\ &CVE\text{:}\, exploits(CVE\text{:}\, \mathtt{threat}_{\mathtt{Inbound192.168.0.0/16SrcIPPkt}}, \\ &\qquad\qquad CVE\text{:}\, \mathtt{vul}_{\mathtt{UnAuthenInbound192.168.0.0/16PktToHost}}) \end{aligned}$$

Principal *RH* can form a chain of trust based on its trust for *NIST* and *NIST*'s trust for *CVE*. As a consequence, *RH* can define a suitable iptables rule (countermeasure) that mitigates the vulnerability of unauthenticated 192.168.0.0/16 subnet packets exploited by spoofed packets of the same source IP range.

$$\begin{aligned} &RH\text{:}\, IptablesRule(RH\text{:}\, \mathtt{iptr}_{\mathtt{DropIn192.168.0.0/16SrcIPPktForwardChain}}), \\ &RH\text{:}\, mitigates(RH\text{:}\, \mathtt{iptr}_{\mathtt{DropIn192.168.0.0/16SrcIPPktForwardChain}}, \\ &\qquad\qquad CVE\text{:}\, \mathtt{vul}_{\mathtt{UnAuthenInbound192.168.0.0/16PktToHost}}) \end{aligned}$$

Principal *ACME* in turn receives the following ABox statements from *RH* for which it has no prior trust relationship.

$$RH\text{:}\,Spoofing(CVE\text{:}\,\texttt{threat}_{\texttt{Inbound192.168.0.0/16SrcIPPkt}}),$$
$$RH\text{:}\,UnAuthPkt(CVE\text{:}\,\texttt{vul}_{\texttt{UnAuthenInbound192.168.0.0/16PktToHost}}),$$
$$RH\text{:}\,IptablesRule(RH\text{:}\,\texttt{iptr}_{\texttt{DropIn192.168.0.0/16SrcIPPktForwardChain}}),$$
$$RH\text{:}\,exploits(CVE\text{:}\,\texttt{threat}_{\texttt{Inbound192.168.0.0/16SrcIPPkt}},\ CVE\text{:}\,\texttt{vul}_{\texttt{UnAuthenInbound192.168.0.0/16PktToHost}}),$$
$$RH\text{:}\,mitigates(RH\text{:}\,\texttt{iptr}_{\texttt{DropIn192.168.0.0/16SrcIPPktForwardChain}},\ CVE\text{:}\,\texttt{vul}_{\texttt{UnAuthenInbound192.168.0.0/16PktToHost}})$$

Principal *ACME* can form a chain of trust based on its trust for *NIST* and *NIST*'s trust for *CVE* and *RH*. For example:

$$CVE\text{:}\,Spoofing \sqsubseteq NIST\text{:}\,Spoofing \sqsubseteq NIST\text{:}\,NIST80041Threat \sqsubseteq ACME\text{:}\,Threat$$
$$CVE\text{:}\,UnAuthPkt \sqsubseteq NIST\text{:}\,UnAuthPkt \sqsubseteq NIST\text{:}\,NIST80041Vul \sqsubseteq ACME\text{:}\,Vulnerability$$
$$RH\text{:}\,IptablesRule \sqsubseteq NIST\text{:}\,NIST80041Countermeasure \sqsubseteq ACME\text{:}\,Countermeasure$$

Use Case 3. This scenario is a variation of use case 2. A company may trust one or more vendors for Semantic Threat Graph ABox statements where each vendor may in turn trust other vendors and/or compliance agencies for ABox Semantic Threat Graph statements. For reasons of space, we do not provide example TBox delegation statements and Abox statements.

7 Related Research

The delegation scheme proposed in this paper is based on managing trust across a model of distributed knowledge-bases. While the model is simple, it closely resembles the OWL-DL approach to modular ontologies [22] using the `owl:imports` constructor with a URI based namespace. Future research will explore representing and reasoning about distributed trust in other modular Description Logic languages such as [14, 36]. The TBox intensional reasoning provided by existing OWL-DL reasoners is relatively scalable, however, extensional reasoning is poor for medium to large-sized ABoxes.

A large body of research results exist on Trust Management and decentralized authorization systems, for example, [9, 13, 16, 27]. However, there has been little consideration regarding how it might be applied to managing trust relationships across knowledge-bases, which is considered in this paper. In [31], a centralised reference ontology is developed to represent trust requirements. Agarwal and Rudolph [5] present an ontology for SPKI/SDSI certificates. In [5] SPKI names are represented as concept names while public keys are represented as individuals. However, once the ontology is constructed any reasoning over delegation chains for the purpose of compliance checking is performed outside of the ontology framework.

Trust Management systems typically describe policy and authorization in terms of discrete permissions and/or assertions. In this paper, authority (about

STGs) is defined in terms of Description Logic concepts. Description Logic has been used to describe and reason about RBAC [18] and XACML policies [26] with subsumption providing role/authorization hierarchies, but do not consider the jurisdiction that principals may have over the ontologies in their local knowledge-bases. Semantic SPKI [6] uses subsumption to define SDSI local name bindings, however an external certificate discovery algorithm implements name reduction. In our paper, public keys are used to uniquely identify principals and their name spaces. Future work will extend this to support SDSI naming, based on the logic described in [21].

The requirements of Distributed Semantic Threat Graphs determined our particular use of Trust Management and effectively corresponds to a conventional compliance check [13]: *for a given delegation network, is a principal trusted for some action?* This check returns a binary answer and we believe that the model could be extended to support forms of quantitative trust, by incorporating KeyNote-style [12] compliance values or weights [11, 19] in the delegation statements. We also assume that trust is monotonic, for example, it is safe to rely on a Semantic Threat Graph delegation chain provided by a vendor since the model does not permit other principals to make conflicting assertions about concepts that originated from the vendor's namespace. Supporting non-monotonic trust, including inter-policy-conflicts such as [10], is non-trivial and effectively requires synchronization of the distributed ABox/TBoxes. Providing support for distributed ontologies is an active research topic [8]. The extent to which these other forms of reasoning over the distributed ontology are applicable, and could be supported by extending our current model, is a topic for future research.

8 Conclusion

In this paper, a trust management approach is proposed to decentralize and delegate knowledge for threats and their mitigation (encoded as Semantic Threat Graphs) to one or more trusted providers. That is, the ability to trust-manage the (delegation) relationships that may exist between the providers.

The ontology-based delegation scheme used subsumption to model the delegation of jurisdiction over (potentially distributed) fragments of a Semantic Threat Graph. This paper extends the model from [20] — which did not consider the possibility that threat catalogues may originate from different trusted providers — to a decentralized trust model.

In this paper, the Semantic Threat Graphs knowledge-bases comprised of knowledge about standards and best practices for threat mitigation using firewalls. The applicability of the (centralised) approach of encoding numerous best-practices is demonstrated in [20]. We argue that the effort of decentralizing this cataloging exercise is comparable. Future work will consider constructing Semantic Threat Graphs from additional threat knowledge-bases such as NVD.

Acknowledgements. The authors would like to thank the anonymous reviewers for their valuable feedback. This research has been supported by Science Foundation Ireland grant 08/SRC/11403.

References

1. http://www.nist.gov/
2. http://www.us-cert.gov/
3. http://www.securityfocus.com
4. Abadi, M., Burrows, M., Lampson, B., Plotkin, G.: A calculus for access control in distributed systems. ACM Trans. Program. Lang. Syst. 15, 706–734 (September 1993), http://doi.acm.org/10.1145/155183.155225
5. Agarwal, S., Rudolph, S.: Semantic Description of Behavior and Trustworthy Credentials of Web Services. 6th International Semantic Web Conference, Busan, Korea (November 2007)
6. Agudo, I., Lopez, J., Montenegro, J.A.: Enabling attribute delegation in ubiquitous environments. Mobile Netw Appl pp. 1–13 (Jul 2008), http://www.springerlink.com/content/q845pp64672m3586/
7. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press (March 2003)
8. Bao, J., Voutsadakis, G., Slutzki, G., Honavar, V.: Modular ontologies. chap. Package-Based Description Logics, pp. 349–371. Springer-Verlag, Berlin, Heidelberg (2009)
9. Becker, M., Fournet, C., Gordon, A.: Design and semantics of a decentralized authorization language. 20th IEEE Computer Security Foundations Symposium (Jan 2007)
10. Bertino, E., Jajodia, S., Samarati, P.: Supporting multiple access control policies in database systems. In: Proceedings of the 1996 IEEE conference on Security and privacy. pp. 94–107. SP'96, IEEE Computer Society, Washington, DC, USA (1996), http://dl.acm.org/citation.cfm?id=1947337.1947353
11. Bistarelli, S., Martinelli, F., Santini, F.: A semantic foundation for trust management languages with weights: An application to the rt family. In: Proceedings of the 5th international conference on Autonomic and Trusted Computing. pp. 481–495. ATC '08, Springer-Verlag, Berlin, Heidelberg (2008)
12. Blaze, M., Feigenbaum, J., Ioannidis, J., Keromytis, A.D.: The keynote trust-management system, version 2 (September 1999)
13. Blaze, M., Feigenbaum, J., Lacy, J.: Decentralized trust management. In: Proceedings of the IEEE Symposium on Research in Security and Privacy. pp. 164–173. IEEE Computer Society Press, Oakland, CA (1996)
14. Borgida, A., Serafini, L.: Distributed Description Logics: Directed Domain Correspondences in Federated Information Sources. On The Move to Meaningful Internet Systems: CoopIS, Doa, and ODBase, LNCS (February 2002)
15. Cuppens-Boulahia, N., Cuppens, F., de Vergara, J.E.L., Guerra, J., Debar, H., Vazquez, E.: An Ontology-Based Approach to React to Network Attacks. 3rd International Conference on Risk and Security of Internet and Systems (CRiSIS), Tozeur, Tunisia (October 2008)
16. Ellison, C., Frantz, B., Lampson, B., Rivest, R.L., Thomas, B., Ylonen, T.: SPKI certificate theory (September 1999)
17. Fenz, S., Goluch, G., Ekelhart, A., Riedl, B., Weippl, E.R.: Information Security Fortification by Ontological Mapping of the ISOIEC 27001 Standard. 13th Pacific Rim International Symposium on Dependable Computing (PRDC), Australia (December 2007)

18. Finin, T., Joshi, A., Kagal, L., Niu, J., Sandhu, R., Winsborough, W.H., Thuraisingham, B.: ROWLBAC - Representing Role Based Access Control in OWL. 13th Symposium on Access control Models and Technologies, Colorado, USA (June 2008)
19. Foley, S.N., Adams, W.M., O'Sullivan, B.: Aggregating trust using triangular norms in the keynote trust management system. In: Cuéllar, J., Lopez, J., Barthe, G., Pretschner, A. (eds.) STM. Lecture Notes in Computer Science, vol. 6710, pp. 100–115. Springer (2010)
20. Foley, S.N., Fitzgerald, W.M.: Management of Security Policy Configuration using a Semantic Threat Graph Approach. Journal of Computer Security (JCS), IOS Press, Volume 19, Number 3 (2011)
21. Foley, S., Abdi, S.: Avoiding delegation subterfuge using linked local permission names. In: Proceedings of 8th International Workshop on Formal Aspects of Security and Trust (FAST2011). LNCS, Springer (2011)
22. Grau, B.C., Horrocks, I., Kazakov, Y., Sattler, U.: Modular Resuse of Ontologies: Theory and Practice. Journal of Artificial Intelligence Research 31 (February 2008)
23. Hernan, S., Lambert, S., Ostwald, T., Shostack, A.: Uncover Security Design Flaws Using The STRIDE Approach. http://microsoft.com/
24. Herzog, A., Shahmehri, N., Duma, C.: An Ontology of Information Security. International Journal of Information Security and Privacy (IJISP), Volume 1, Issue 4 (2007)
25. Kodeswaran, P.A., Kodeswaran, S.B., Joshi, A., Finin, T.: Enforcing Security in Semantics Driven Policy Based Networks. 24th International Conference on Data Engineering Workshops, Secure Semantic Web, Cancun, Mexico (April 2008)
26. Kolovski, V., Hendler, J., Parsia, B.: Analyzing web access control policies. In: Proceedings of the 16th international conference on World Wide Web. pp. 677–686. WWW '07, ACM, New York, NY, USA (2007), http://doi.acm.org/10.1145/1242572.1242664
27. Li, N., Winsborough, W., Mitchell, J.: Distributed credential chain discovery in trustmanagement. Journal of Computer Security 11(3), 35–86 (2003)
28. Ray, I., Poolsappasit, N.: Using Attack Trees to Identify Malicious Attacks from Authorized Insiders. 10th European Symposium on Research in Computer Security, Milan, Italy (September 2005)
29. Schneier, B.: Secrets and Lies Digital Security in Networked World. Wiley Publishing (2004)
30. Smith, M.K., Welty, C., McGuinness, D.L.: OWL Web Ontology Language Guide. W3C Recommendation, Technical Report (2004)
31. Squicciarini, A.C., Bertino, E., Ferrari, E., Ray, I.: Achieving Privacy in Trust Negotiations with an Ontology-Based Approach. IEEE Transactions on Dependable and Secure Computing, Volume 3, Issue 1 (2006)
32. Stevens, R.: Unix Network Programming, Networking API's: Sockets and XTI, 2nd Edition, Volume 1. Prentice Hall (1998)
33. Thuraisingham, B.: Building Trustworthy Semantic Webs. AUERBACH (2007)
34. Tracy, M., Jansen, W., Scarfone, K., Winograd, T.: Guidelines on Securing Public Web Servers: Recommendations of the National Institute of Standards and Technology. NIST Special Publication 800-44, Version 2 (September 2009)
35. Wack, J., Cutler, K., Pole, J.: Guidelines on Firewalls and Firewall Policy: Recommendations of the National Institute of Standards and Technology. NIST-800-41 (2002)
36. Wang, Y., Haase, P., Bao, J.: A survey of formalisms for modular ontologies. In: International Joint Conference on Artificial Intelligence (IJCAI'07) Workshop (2007)