COMPUTERS & SECURITY 67 (2017) 35–58

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Data-driven analytics for cyber-threat intelligence and information sharing

Sara Qamar [a], Zahid Anwar [a,b,*], Mohammad Ashiqur Rahman [c], Ehab Al-Shaer [d], Bei-Tseng Chu [d]

[a] School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad, Pakistan
[b] Department of Math and Computer Science, Fontbonne University, St. Louis, USA
[c] Department of Computer Science, Tennessee Tech University, Cookeville, TN, USA
[d] Department of Software and Information Systems, University of North Carolina at Charlotte, Charlotte, USA

ARTICLE INFO

Article history:
Received 22 February 2016
Received in revised form 20 November 2016
Accepted 6 February 2017
Available online 10 February 2017

Keywords:
Ontology
STIX
Network
Risk analysis
Impact
Reachability
Cyber threat intelligence
Attribution
Reasoning
OWL

ABSTRACT

Efficient analysis of shared Cyber Threat Intelligence (CTI) information is crucial for network risk assessment and security hardening. There is a growing interest in implementing a proactive line of defense through threat profiling. However, determining the resiliency of a particular network with respect to relevant threats reported in CTI shared data remains a challenge, largely due to the lack of semantics and contextual information present in textual representations of the threat knowledge. To overcome the limitations of existing CTI frameworks, we devise a threat analytics framework based on Web Ontology Language (OWL) for formal specification, semantic reasoning, and contextual analysis, allowing the derivation of network associated threats from large volumes of shared threat feeds. Our ontology represents constructs of Structured Threat Information eXpression (STIX) with the additional concepts of Cyber Observable eXpression (CybOX), network configurations, and Common Vulnerabilities and Exposure (CVE) for risk analysis and threat actor profiling. The framework provides an automated mechanism to investigate cyber threats targeting the network under question by classifying the threat relevance, determining threat likelihood, identifying the affected and exposed assets through formulated rules and inferences. We perform a comprehensive structural and conceptual evaluation of critical advanced persistent threats (APTs) collected from credible sources and determine their relevance and risk posed to realistic network case studies. Finally we show that the proposed framework is novel in the type of analytics it provides and outperforms other competing approaches in terms of efficiency and effectiveness.

1. Introduction

The exponential increase in cyber-attacks with the proliferation of sophisticated hacking tactics is creating strong security concerns for network administrators and users. It is a need of the hour to automate intelligence gathering especially for risk assessment. The traditional approach of manually identifying, categorizing, and then countering each threat is not effective when dealing with a diversified and voluminous set of attack vectors in the form of advanced persistent threats (APTs). Sharing of threat information between various communities via Cyber Threat Intelligence (CTI) (Sans, 2015) frameworks has been recently gaining momentum with the intent of creating a proactive

* Corresponding author.
E-mail addresses: 13msccssqamar@seecs.nust.edu.pk (S. Qamar), zahid.anwar@seecs.nust.edu.pk, zanwar@fontbonne.edu (Z. Anwar), marahman@tntech.edu (M.A. Rahman), ealshaer@uncc.edu (E. Al-Shaer), billchu@uncc.edu (B.-T. Chu).
http://dx.doi.org/10.1016/j.cose.2017.02.005
0167-4048/

line of defense based on knowledge of impending attacks and understanding of attackers' intentions and capabilities. CTI is a collective evidence based on information gathered from multiple sources and it focuses on actionable response toward emerging threats. These intelligence data also assist in monitoring and maintaining threat profiles. Threat profiling provides detailed knowledge regarding threat campaigns, victims, motivation, tools, and methodologies that are followed to attack a network. Threat profiles help network administrators and cyber analysts to take decisions to defend the network and mitigate threats proactively.

Structured Threat Information eXpression (STIX) (MITRE, 2013b) is one such community-driven effort to develop a standardized language to define cyber threats and document their instances reported at different collaborating nodes. With the sharing of threat indicators, it also specifies ways to manage information related to threat actors, vulnerabilities being exploited, tactics, techniques, and procedures of attacks, incident location, and associated campaign, and finally determines the courses of actions. STIX is an umbrella standard that builds on top of MAEC (MITRE, 2015a), CAPEC (MITRE, 2015b), TAXII (MITRE, 2015c), and CYBOX (MITRE, 2015d). The information recorded using STIX is periodically shared among trusted parties using Trusted Automated eXchange of Indicator Information (TAXII) (MITRE, 2015c), which provides enhanced situational awareness regarding emerging threats with the intention that they will help in their timely and efficient neutralization. There exist several threat discovery services (Bob Gourley, 2014) that focus on information gathering and sharing regarding threat incidents, indicators, and attack procedures. The information gathered from various sources is critical for cyber defense and needs to be normalized and validated. For example, Financial Services Information Sharing and Analysis Center (FS-ISAC, 2015) and Research and Engineering Networking Information Sharing and Analysis Center (REN-ISAC, 2008) regularly maintain and distribute threat intelligence data for financial industry members around the globe. Similarly, Hail a Taxii (2015) works in collaboration with different communities, providing CTI data as a free service with a current size of threat indicators amounting to over half a million. Considering the volume, diversity, and complexity of the information reported by such services, manual threat analytics of these feeds is simply impractical.

Usually, STIX reports are generated manually by the security analysts. Limited mechanisms are available that can validate the STIX reports before sharing. As a result, shared CTI data often include incomplete or incorrect information. Moreover, due to the obscure nature of security data, new or sometimes existing data about a single attack tend to be reported across different threat reports as security analysts independently examine and report various instantiations of the same attack. This non-uniformity, as well as redundancy of data, makes it more challenging to analyze a sample CTI report for identifying its relevance for a particular network. The sole purpose of TAXII is to exchange STIX among communities and, thus, the millions of shared STIX using TAXII is not necessarily relevant to particular network configurations. The sharing mechanism does not determine the relevance of reported threats with a network. Therefore, an automated, efficient, and usable analytics derived directly from the data is greatly needed for the aspired utilization of CTI sharing.

This work addresses this need and proposes a framework to perform analytics on data obtained from existing repositories of intelligence frameworks (STIX/TAXII) to identify threat relevance on a network. We term this framework *STIX-Analyzer*. The proposed methodology works by defining OWL-based ontology for network, Common Vulnerabilities and Exposures (CVEs) and STIX. The term "ontology" stands for a mathematical, logical, and machine readable model with semantic meaning. OWL is highly expressive that allows automatic inference and semantic reasoning based on defined domain knowledge and concepts (Stumptner et al., 1998; Yang et al., 2009). OWL holds two types of properties: (i) object and (ii) data. Object properties relate a member with other members while data properties relate a member to data. Ontology data types are the data values for instances and object properties are the links between instances. The ontology restrictions are the associations that must hold for all instances of a class. Restrictions such as "equals", "some" (some values from a range), "has" (at-least one value), and "at most" are used to hold relationships between the members of a class. Existential ("some") restrictions are more frequently used in a designed ontology to quantify instances of a class that are connected with other instances via some or at least one property to hold relationship.

We populate our ontology model with elements extracted from descriptions of emerging threats, such as Tactics Techniques and Procedures (TTPs), indicators, observables, exploit targets as well as from CVEs. The network ontology is designed to analyze threat data on network components. The information provided by STIX is then used to identify vulnerabilities and associated risk present in the targeted network. We employ logic based deductive inference rules defined in Semantic Web Rule Language (SWRL) (W3C, 2004) that operate on our ontology model and perform mapping of the threat, vulnerabilities and network elements. We have chosen the Protege 3.5 (Stanford, 2014) OWL editor because it is an open source tool with extensive community support and features. Furthermore we have selected the OWL-Manchester syntax (Horridge et al., 2006) because it is minimally verbose, user friendly, and less logician as compared to other syntaxes for the OWL design and to represent ontological concepts. The Manchester syntax has become the default syntax in Protege and is also supported by ontology editors like TopBraid Composer (TopQuadrant, Inc, 2001). Different reasoners are also available. We preferred to use the open-source Java based reasoner called Pellet (Complexible/pellet, 2013) because of its provision for SWRL built-in rule development and execution. Pellet also helps in verifying the consistency of ontology classes and instances hierarchy. Protege provides the *SWRLDroolsTab* as a *Drools rule engine* (Protegeproject, 2014) that executes SWRL rules in Protege and provides updates for inferred values.

STIX-Analyzer improves the capability of timely identification of threats and corresponding risk on a network by aiming for automated, dynamic, and actionable intelligence, contrary to the traditional manual analysis of threats. It is a novel approach that presents a comprehensive semantic model considering STIX, Network, and CVE altogether to relate the shared threats knowledge with network architectural schematics and analyze the impact of potential threats and attacks on the network. The impact derivation is based on threat likelihood, total loss of affected assets, and threat reachability to network.

STIX-Analyzer also maintains profiles of threats. Several high-level use cases (UC) are supported (MITRE, 2013c) by the current version of STIX which are as follows: (UC-1) analyzing cyber threats, (UC-2) specifying indicator patterns for cyber threats, (UC-3) managing cyber threat response activities, (UC-4) situational awareness, (UC-5) managing content over time and the (UC-6) sharing of information. These use cases in turn constitute multiple low level use cases. The proposed STIX-Analyzer framework covers several of these through its automated analysis process. *STIX-Analyzer* directly addresses UC-1 based on its threat actor characterization (Section 4), campaign analysis (Section 4.2), incident analysis by measuring *hasAssetsRelevance* and *hasImpactRelevance*, exploit target analysis using *hasCveRelevance*, asset risk analysis by measuring total loss of affected assets, and finally victim targeting analysis using *hasOrganizationRelevance*, *hasTargetedLocationRelevance* and *hasTargetedLanguageRelevance* (Section 3.4). UC-3 is also directly addressed using prioritization of cyber threats through threat likelihood and *hasMotivationRelevance* derivation. Finally, the intrinsic advantages of using an ontology (OWL) driven system such as gaining the ability to capture context, descriptional logic based consistency checking, extensibility and ease of sharing indirectly address UC-4, UC-5 and UC-6. The reasoning process is independent of the underlying procedures used to maintain repositories for threats, network, and vulnerabilities. A comprehensive evaluation, considering both efficiency and effectiveness criteria, is performed on the proposed analyzer. The criteria to measure effectiveness include ontology structure, clarity, consistency, accuracy of the analysis and feature novelty. The efficiency criterion is based on performance evaluation and is measured in terms of processing time, resource utilization, usability, and ease of configuration.

The remaining paper is organized as follows. Section 2 includes the literature review of the proposed work. In Section 3, we present the architecture of the *STIX-Analyzer* framework and proposed methodology in detail. Section 4 elaborates the mechanism for attack attribution. The proposed framework is evaluated in Section 5. We conclude this paper with future directions in Section 6.

2. Related work

There are several recently launched commercial CTI endeavors describing structured and unstructured threat information. Tsai and Chan (2007) analyzed cyber threat intelligence at a time when standards for representing threat information were non existent. Before the emergence of CTI standards, security experts and researchers posted their cyber security threat findings and observations on web blogs. The authors analyzed Weblog posts for various categories of cyber security threats related to the detection of cyber-attacks, crime, and terrorism. Latent Semantic analysis (LSA) (Landauer et al., 1998) was used to find semantically related topics in web blog corpus. Further important keywords of each topic are assigned quantitative measure through Probabilistic LSA (PLSA) (Hofmann, 1999). The results prove the capability of this approach for broadly searching security related news in massive web blogs. The usefulness of this approach is limited because of the limitation of web blogs in representing threat scenarios in a fine-grained, real-time, and uniform manner. The Public Regional Information Security Event Management (PRISEM) system (PRISEM – Office of the Chief Information Officer, 2011) performs threat monitoring that generates warning for cyber-attacks based on security and information event management (SIEM). This system is intended for maintaining logs regarding cyber events and it lacks threat analytics and impact derivation abilities. Li et al. (2007) focus on the problem of true threat identification in a distributed environment where network security data are managed at distributed locations. The proposed approach provides the means of finding correlations between alerts arriving from distributed components. The major limitation of this work is the lack of standardization as alert data need to be converted to a uniform representation as devices and tools are from heterogeneous vendors.

ThreatConnect (2012) offers a Threat Intelligence Platform (TIP) which collects data from multiple sources to perform analysis and looks for indicators and their associations with other entities such as adversaries, signatures and incidents. ThreatStream OPTIC is a cyber-threat intelligence platform, which analyzes threats from different sources, ranks each indicator and defines relationships with known threats (Threatstream, 2014). ThreatQ is a threat intelligence platform (TIP) (ThreatQuotient, 2015) that manages attack related data in a central repository and prioritizes threats. Its objective is to provide users with integrated intelligence correlated from multiple sources. CISCO is developing techniques (McGrew and Verma, 2015) for recommending actions based on STIX reports. These techniques rely on security information and event management (SIEM) systems to analyze and monitor detected threats. The user has to approve or select a course of action (block, capture, prioritize, etc.), and then proceed with monitoring and repeat the same procedure until the issue gets resolved.

Extensions to the STIX language itself for improving analytics are also being studied by a few communities. For example, Fransen et al. (2015) analyze the timely gain of information regarding incidents from cyber security information sharing, proposing STIX for cyber situational awareness and utilizing its vocabulary to enumerate impact regarding data. Burger et al. (2014) present a detailed layered taxonomy model to analyze CTI exchange and classifying cyber security terms. A STIX-based ontology and the need for STIX automation are discussed in Visitology (2014). The article states that due to the complexity of STIX reports and the issue of interoperability, it is not trivial to directly use STIX feeds in threat analytics. However, these endeavors to analyze STIX data do not comprehensively address the problem of performing (formal) inference or relevance analysis with respect to network mapping, risk assessment, and threat attribution.

3. Proposed methodology

In this section, we present the *STIX-Analyzer* framework. Fig. 1 shows the high-level architecture of this framework.

3.1. *Defining ontologies for STIX, network, and CVE*

The ontology for the threat analytics is defined by assembling information from three different sources: STIX (xsd)

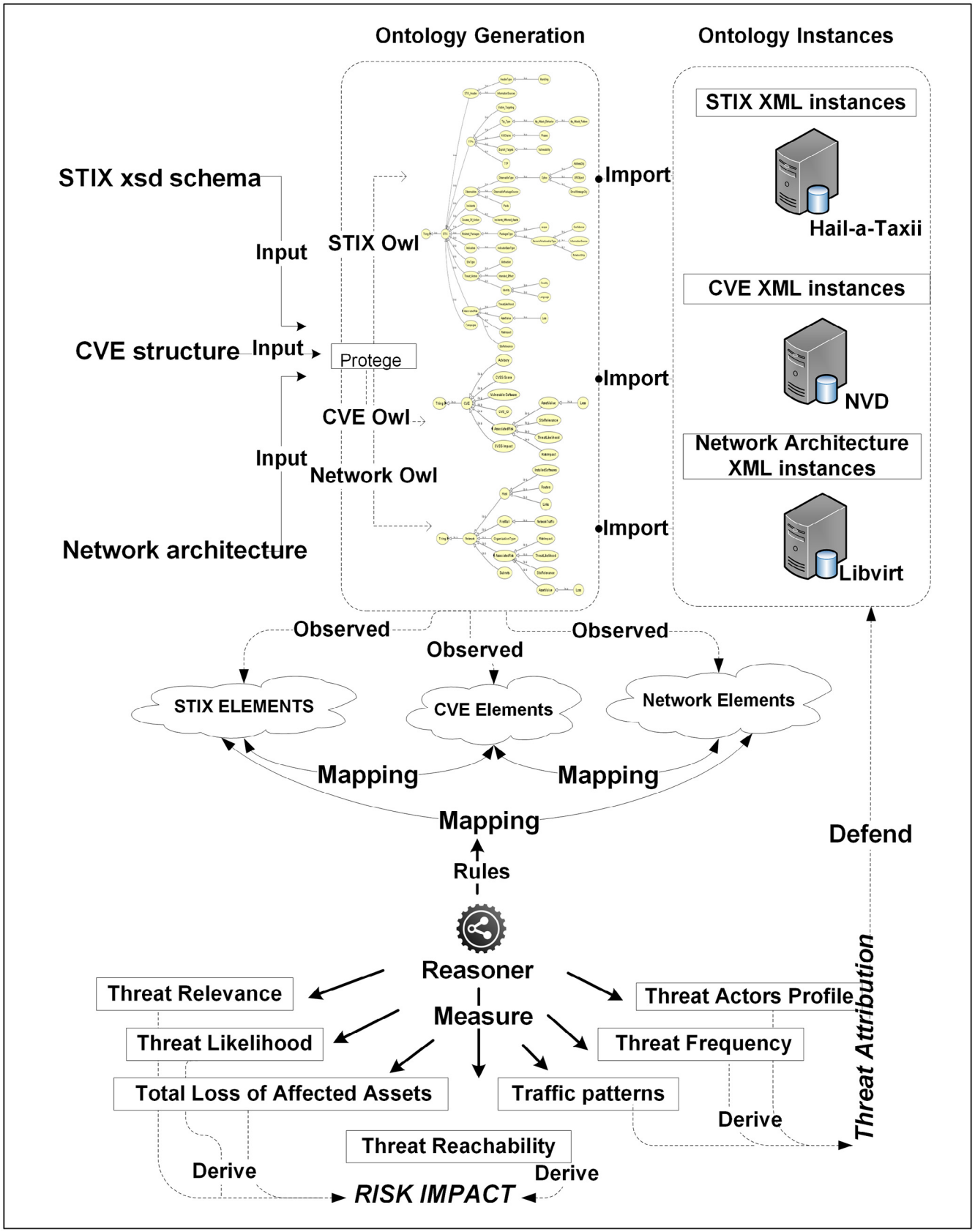


Fig. 1 – STIX-Analyzer.

schema document, CVE description along with identifiers, and the network architecture model. The STIX schema document contains xml elements and attributes that represent threat knowledge (MITRE, 2015e). The National Vulnerability Database (NVD) maintains CVE descriptions and consequently a Common Vulnerability Scoring System (CVSS) to score vulnerabilities (NVD, 2015b). The network architecture consists of various entities, such as subnets, firewalls, hosts, and links among connected hosts. We have followed the OWL-Manchester syntax to develop the ontology. The ontology (OWL) is developed by creating classes, object and data properties, according to the gathered information, using Protege (CO-ODE Project, 2009; Knublauch et al., 2004). The ontology classes are further divided into sub-classes having properly defined domains and ranges for each property which are used to define relations. After the design step, instances (also known as individuals) are imported in order to populate the developed ontology model.

All ontologies have a root class *Thing* by default and all classes are derived from *Thing*. Thus, the proposed threat analytics ontology model is also rooted at *Thing*, and the model components (STIX, CVE, and Network) are inherited from it. Fig. 2 shows the high level ontology class view. The relations among the different properties of these three major components and a few inferred properties from the designed rules are shown in Fig. 3.

STIX ontology:

A comprehensive ontology for STIX is built as proposed in its schema documents (MITRE, 2013d). The proposed concepts of STIX ontology comprise *Observables*, *Indicators*, *Incidents*, *TTPs*, *ExploitTargets*, *Campaigns*, *ThreatActors*, and *CourseOfActions*. These concepts are declared as separate classes in the STIX ontology and each class is further divided into sub-classes, data properties, and object properties. As a way of illustration and due to space limitations Fig. 2 shows a zoomed in portion of Fig. 1 illustrating some crucial classes of the STIX owl. The important classes and properties are briefly described below.

- **STIXHeader** includes a sub-class *HeaderType*, and data properties: *id* which is unique for each shared threat and a *timestamp* referencing the threat description.
- **Observables** describes STIX observables with the help of a sub-class *ObservableType* that specifies *Cybox* which is cyber observable expression language using the *hasProperties* relation.
- **ThreatActors** specifies information about a threat source's identity, motivation, and its intended effect. Its sub-classes include *Identity*, *IntendedEffect*, and *Motivation*.
- **CourseOfActions** characterizes a course of action that may be taken in response to an attack or as a preventive measure prior to an attack. It does this with the help of sub-classes that detail the *Stage*, *Efficacy*, *Cost*, and *Impact*.
- **TTPs** includes the tactics, techniques, and procedures used to launch an attack, such as the phase in the "kill-chain" intrusion model, malware behavior, and the victim. *TTPs* has multiple *TTP* that further comprises *ExploitTargets*, *VictimTargeting*, and *Behavior*.
- **ExploitTargets** represents vulnerabilities using cve ids.
- **Campaigns** attributes a threat to a particular actor using the sub-classes: *Attribution* and *Activites*.
- **Incidents** has information regarding the *Victim*, status of *SecurityCompromise*, *IncidentsAffectedAssets*, and *ImpactAssessment* of reported threats.
- **Indicators** describes patterns for the observed attack using *Cybox*, and has data properties for *Hashes* and *Signatures*.

STIX Ontology Relations and Restrictions. The relations between various concepts are mapped in the ontology by defining properties and specifying their domains, ranges, and restrictions. For instance, Listing 1 shows the restrictions of the *hasObservable* object property, expressed in the OWL xml concrete syntax where the domain of *hasObservable* is *Indicators* and its range is *Observables*. Listing 2 shows that *Indicator* is a subclass of STIX and it has a property *hasObservable* that restricts the instances of the indicator to have at least one instance of type *Observable* using the restriction property of *has* (∈). Properties *hasIndicator* and *hasTTP* allow the instances of *Indicators* to be associated with one or more instances of type *Indicators* and *TTPs* using existential *some* (∃) property restriction. An indicator can hold any of these properties which is depicted by the symbol (∪) expressing the Union relationship.

```
<owl:ObjectProperty rdf:about = "\#hasObservable">
    <rdfs:range rdf:resource = "\#Observables"/>
    <rdfs:domain rdf:resource = "\#Indicators"/>
</owl:ObjectProperty>
```

Listing 1: STIX Object Property in OWL XML Syntax

```
Indicator ≡ STIX ∪
        ∈ hasObservable has Observables ∪
        ∃ hasIndicator some Indicators ∪
        ∃ hasTTP some TTPs
```

Listing 2: STIX Indicator Restrictions as DL Rules

Network ontology

The ontology for the network consists of hosts, firewalls, routers, subnets, links, and their properties, including versions of installed software, operating system, configured hardware, and environment. This modeling is used to demonstrate and analyze the behavior and impact of threats on various network architectures.

Network Ontology Relations and Restrictions: Listings 3 and 4 present a part of the network ontology model. Network ontology *Host* is a subclass of *Network* and has one or more *Firewall* configured (e.g., using *hasFirewall*, as shown in Listing 3). *Host* must be connected to at least one *Router* and linked with other hosts via a router or switch using *hasConnectedHost* property, containing a *Name*, an *IP*, and a *SubnetID* using *min* (≥) restrictions. Some restrictions on a firewall are given in Listing 4. *Firewall* is a subclass of *Network* and generates alerts against *ThreatActors* mapped to the ontology using *hasAlert* property with existential restrictions. A firewall possesses *hasAccess* to *NetworkTraffic*. Some restrictions are used to identify the reachability of threats to network hosts by analyzing *hasHostSrc*, *hasHostDest*, and *hasNextHop* and then access is allowed through the firewall. Restrictions force the firewall to be configured for at least one source, one destination, and the next hop (to forward the packet) using *min* restrictions.

```
Host  ≡ Network ∪
        ∃   hasFirewall some Firewall ∪
      ≥   hasRouter min 1 ∪
      ≥   hasConnectedHost some Host ∪
      ≥   hasHostName min 1 ∪
      ≥   hasHostIP min 1 ∪
```

Listing 3: Network Host Restrictions as DL Rules

CVE ontology

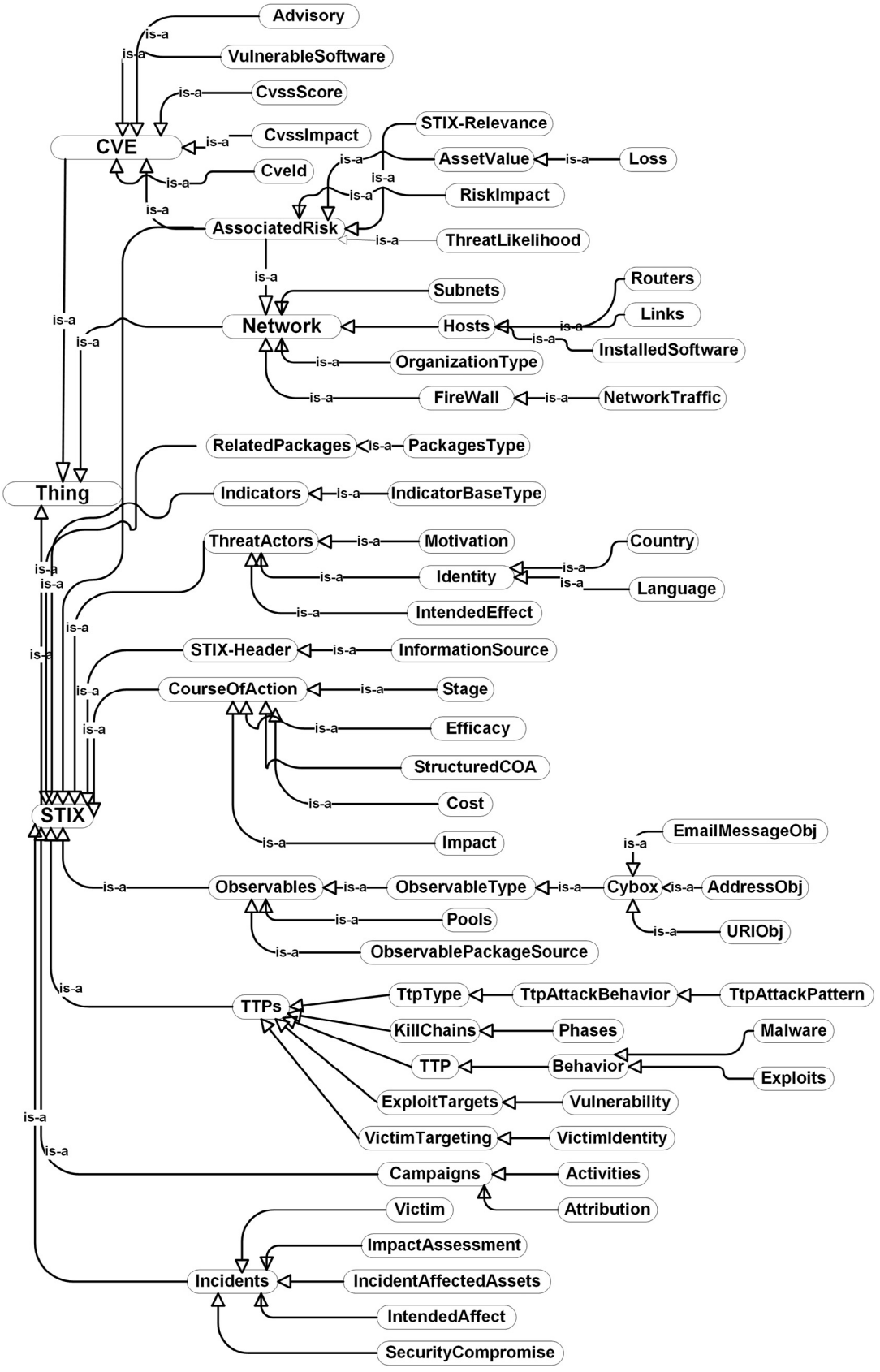


Fig. 2 – STIX-Analyzer Ontology.

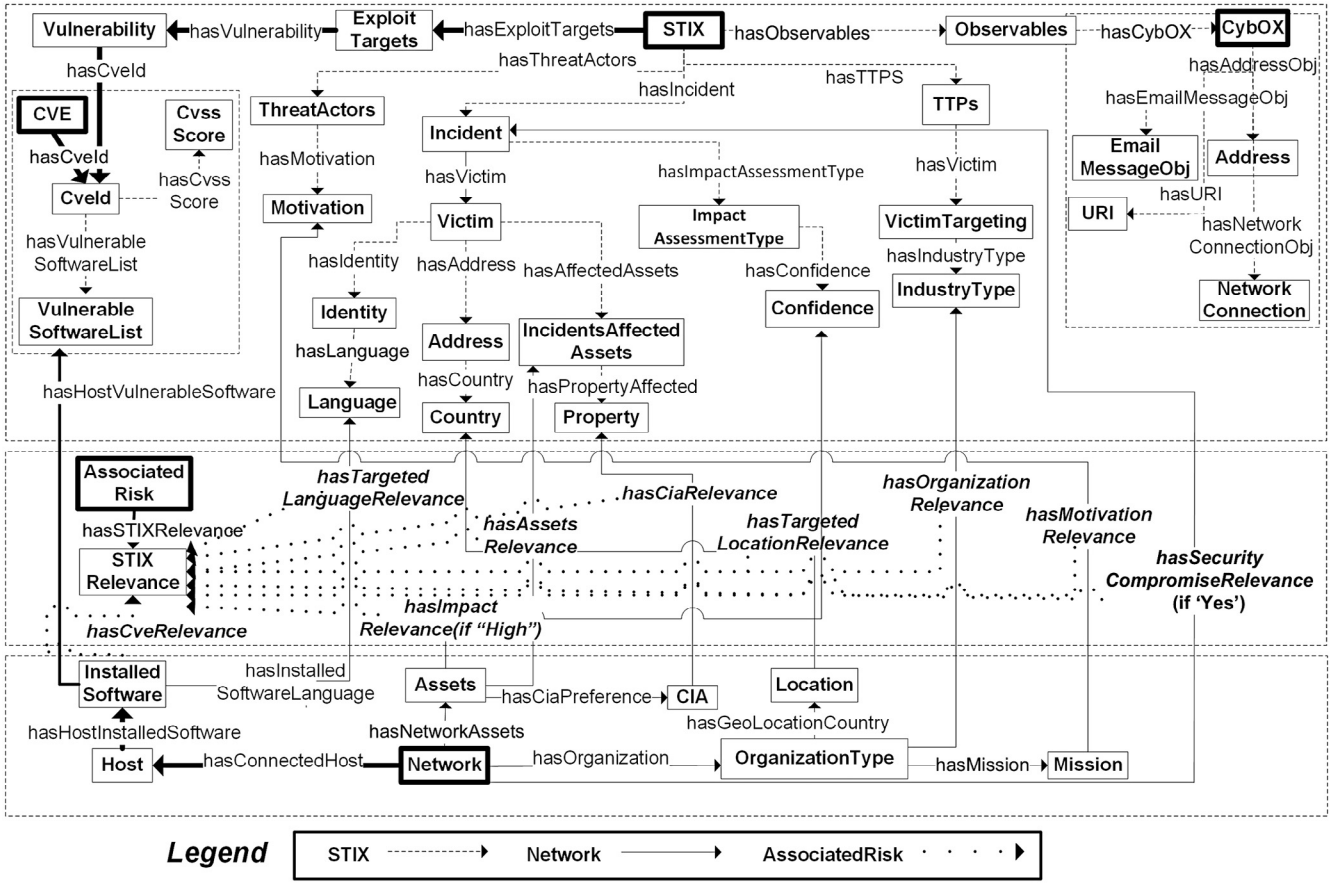


Fig. 3 – Ontological model relations.

```
Firewall ≡ Network ∪
        ∃ hasAlerts some ThreatActors ∪
    ∃ hasAccess some NetworkTraffic ∪
    ≥ hasHostSrc min 1 ∪
    ≥ hasHostDest min 1 ∪
    ≥ hasNextHop min 1
```

Listing 4: Network Firewall Restrictions as DL Rules

The design of the CVE ontology is based on the NVD scoring system for known vulnerabilities (NVD, 2015b). The CVE class has *CveId*, *VulnerableSoftware*, *CvssScore*, *Advisory*, and *Impact* sub-classes.

CVE Ontology Relations and Restrictions. A few restrictions are mapped to the *CVE* class as shown in Listing 5. The CVE class is derived from the root *owl Thing* class. A single instance of *CVE* contains at most one cve id, a cvss score, and an impact value as illustrated in Listing 5, using *max* restriction. CVE also includes *hasVulnerableSoftwareList*, *hasAdvisory* data properties, which are modeled using at least one (*min*) restriction.

```
CVE ≡ owl:Thing ∪
        ≤ hasCveId max 1 ∪
    ≤ hasCvssBaseScore max 1 ∪
    ≤ hasImpactSubscore max 1 ∪
    ≥ hasVulnerableSoftwareList min 1 ∪
    ≥ hasAdvisory min 1
```

Listing 5: CVE Restrictions as DL Rules

Associated Risk

The *AssociatedRisk* class is derived from STIX, Network, and CVE classes. This class carries the derived result from the reasoning performed for *Impact* and *RelevanceScore* computation through rules, as discussed in the subsequent sections. The *AssociatedRisk* class has sub-classes: *RiskImpact*, *StixRelevance*, *ThreatLikelihood*, *Vulnerabilities*, and *AssetValues*. Some properties are used to compute the STIX reports' relevance with the network. The relevance results are derived by executing vulnerability identification, software relevance, attacker's motivation, victim's business type, targeted location, affected assets, CIA relevance, incident security relevance, and victim's language relevance rules. The derived results of these rules are stored in data properties of the *AssociatedRisk* class. A high level view of the mapped relations that are used to compute rules and derive results is provided in Fig. 3. As a reference, we have highlighted some properties used in Listing 9 to compute *STIX* relevance with the network on the basis of vulnerable software in Fig. 3. STIX *ExploitTargets* contains vulnerabilities or cve ids shown at the top of the figure, where STIX *ExploitTargets* class is linked to the *Vulnerability* class using the *hasVulnerability* object property. *Vulnerability* is connected to *CveId* class. *CveId* is further linked to *VulnerableSoftwareList* (list of vulnerable software products) using *hasVulnerableSoftwareList* relation. Network *Hosts* are connected to the network via a router or switch using the *hasConnectedHost* relation. The *InstalledSoftware* class for installed software on the host is linked using the *hasHostInstalledSoftware* relation. *Host hasHostInstalledSoftware* is compared with CVE *hasVulnerableSoftwareList* to identify vulnerable software installed on the network hosts.

3.2. *Imported instances*

In order to perform meaningful threat to network mapping and analytics, the designed ontology needs to be automatically populated with real instances of network and STIX, along with the vulnerability information obtained from CVEs. STIX feeds and various threat incident reports are imported into *STIX-Analyzer's* ontology model. These STIX feeds are made available by open CTI repositories and threat incident reports posted by security vendors (NVD, 2015b; STIXProject/schemas-test, 2015; TAXII, 2015). A text mining parser (depicted in Algorithm 1) allows *STIX-Analyzer* to scan textual APT reports and transform the knowledge into STIX (xml) instances. For example STIX *ExploitTargets* as shown in Listing 6 identifies various *Vulnerabilities* corresponding to *CveId*.

Threat reports are supplied as input to the parser which uses a combination of regular expression matching and natural language processing (NLP) techniques to extract keywords pertaining to relevance factors based on the STIX vocabulary (MITRE, 2013d; OASIS, 2015). Regular expression matching allows extraction of vulnerabilities based on CVE Ids. The geolocation library allows the identification of threat locations mentioned in the reports. NLP techniques like stemming and synonym enumeration help to parse contextually similar keywords related to properties, such as targeted assets, type of security compromise, and impact.

Similarly *STIX-Analyzer* employs the network architectural knowledge represented in xml format for analyzing threats. For example, some elements from an imported network instance are shown in Listing 7. We imported sample network

Algorithm 1 STIX Parser Scans For Relevance Factors in Threat Reports

Input : *Threat_Report, AssetsTypeVocab, MotivationVocab, IntendedEffectVocab, LossPropertyVocab, LanguageDict, CIQ_IndustryType*
Output: *cve_list, motivation_list, organization_list, location_list, assets_list, language_list, cia_list, effected_list*

```
for each word in Threat_Report : do
    ▹ Find list of cve ids in Threat report using cve regex.
    match (cve_regex in word )
    if matched then
        cve_list ← word
    end

    ▹ Find motivation in Threat report using motivation vocabulary.
    if ! matched then
        for each vocab in MotivationVocab : do
            match (vocab in word)
            if matched then
                motivation_list ← word
            else
                ▹ Find synonyms of motivation vocabulary in Threat
                match (synonyms(vocab) in word )
                if matched then
                    motivation_list ← word
                else
                    ▹ Find stemming words and stemmed synonyms of motivation vocabulary in Threat
                    match (stemming(vocab) OR stemming (synonyms(vocab)) in word)
                    if matched then
                        motivation_list ← word
                    end
                end
            end
        end
    end
    Repeat the second if Steps to :
    a. Find assets_list in Threat report using AssetsTypeVocab .
    b. Find organization_list in Threat report using CIQ_IndustryType .
    c. Find cia_list in Threat report using LossPropertyVocab .
    d. Find location_list in Threat report using GeoLocation .
    e. Find language_list in Threat report using LanguageDict .
    f. Find effected_list in Threat report using IntendedEffectVocab .
end
```

```
<STIX:ExploitTargets>
        <Vulnerability>
        <CveId> CVE-2009-3129 </CveId>
    </Vulnerability>
    <Vulnerability>
        <CveId> CVE-2010-3333 </CveId>
    </Vulnerability>
    .....
</STIX:ExploitTargets>
```

Listing 6: STIX ExploitTargets as XML Instance

topology instances (generated from BRITE Topology Generator (NSNAM, 2011) and Libvirt Virtualization API (libvirt, 2015)) into our ontology model. We preferred BRITE as it is a universal topology generator (Medina et al., 2001) that supports hierarchical topologies, flat router, and is flexible for configuration of several network parameters. Although BRITE is no

longer supported by its developers, it has been proven to be useful to model large scale and real-world networks for threat analysis to map different network scenarios (Heckmann et al., 2003; Zheng et al., 2012).

```
<Network>
    <hasHostName> Host-A </hasHostName>
    <hasHostInstalledSoftware>
        Microsoft Office 2003 SP3
    </hasHostInstalledSoftware>
<Network>
```

Listing 7: Network Host as XML Instance

Vulnerability information in the form of xml formatted CVE and CVSS scores from the NVD database (NVD, 2015a) is also imported into the ontology model as *CVE* class instances to identify network related vulnerabilities. A CVE instance, as shown in Listing 8, has a *CveId* of CVE-2010-3333, *CvssScore* of 10 and a *VulnerableSoftwareList*. The latter includes Microsoft Office 2003 SP3 and Microsoft Office 2007 SP2. Prior to the instance import process, minor cleaning needs to be performed on the STIX, network, and CVE documents when they are downloaded from their respective repositories for removal of supplementary xml tags like version of STIX used, schema description, and vendor details.

```
<CVE>
    <CveId> CVE-2010-3333 </CveId>
    <CvssScore> 10 </CvssScore>
    <VulnerableSoftwareList>
        <Product>
        Microsoft Office 2003 SP3
        </Product>
        <Product>
        Microsoft Office 2007 SP2
        </Product>
    </VulnerableSoftwareList>
    ...........
<CVE>
```

Listing 8: CVE as XML Instance

3.3. *Mapping threat to network*

STIX provides a provision for capturing extensive knowledge of particular threats. The STIX schema is highly complex comprising of hundreds of data types and properties. Similarly network architectural knowledge is also quite vast with a medium sized network typically comprising of thousands of network entitles. Sifting through relevant elements of STIX that are comparable to the network architecture is a challenging task. We use the benefits of the ontology reasoning process to identify nine factors or attributes (*F*) in STIX and the network that are comparable, and can be used in identifying the potential *Impact* of threats on the network. STIX has *ExploitTargets* and *Vulnerability*, which help in identifying the vulnerable and exploitable software (*hasHostVulnerableSoftware*) installed on network hosts. The threat actor's *hasMotivation* of STIX is mapped to the *hasIntent* of the network. Similarly, the *hasIndustryType* of STIX is matched with *hasOrganizationType* of the network. STIX *hasCountry* is comparable to *hasGeolocationCountry* of the network. If *hasAffectedAssets* of STIX is the same as *hasAssets* of the network, then the network assets are vulnerable to that threat. Some threats target users who belong to a particular nationality, culture or use a specific language. STIX element *hasLanguage* is mapped to *hasSoftwareInstallationLanguage* of the network. *hasPropertyAffected* of STIX and *hasCiaPreference* of the network are comparable and, thus, can be matched to identify relevant threats. *hasSecurityCompromise* of STIX helps in identifying the critical and manifested threats. The associated rules are defined in Section 3.4.

3.4. *Risk analysis*

Once the mapping between threats and network is determined, a number of interesting risk analysis questions can be posed such as: (1) "is my network vulnerable to this threat?" or (2) "if this threat were to manifest what potential losses can occur?" The core of *STIX-Analyzer*'s analytics engine is based on a set of horn-clause style deductive inference rules that operate over the ontology. The notations that we assign to several major concept variables, used in defining these rules, are summarized in Table 1. Within the risk analysis, we derive the impact, denoted as **I**, with the help of four parameters. We term them as four *T*s of the threat analysis, which are as follows: (i) *Threat relevance*, (ii) *Threat likelihood*, (iii) *Total loss of affected assets*, and (iv) *Threat reachability*. These four *T*s are expressed using **F**, **L**, $\bar{\mathbf{A}}$, and **R**, respectively. Each of them is associated with a set of rules. Threat relevance depends on a number of factors (*F*) that we discuss in the following section. We weigh each factor with a relevance score denoted as S_i. Threat likelihood (*L*) depends on these relevance scores which are measured using the various factors. These factors influence the likelihood of a threat occurrence. For instance, if the chance of fulfilling the motivation behind launching attacks on a particular network is high, or if the network lies in a targeted country and the nature of organization matches the type of attackers' target, the likelihood increases. The likelihood of assets being exposed increases when their geographic proximity to other attack victims increases owing to the principle of transitive risk. We calculate loss ($\bar{\mathbf{A}}$) both quantitatively and qualitatively, denoted as A_n and A_l, respectively. Finally, reachability (*R*) helps to measure the number of hosts that are potential targets or victims of the threat incident and analyzes how interconnected network assets are in a transitive-risk relationship with each other, expressed using Eq. (1). A threat vector initiated from host X in network (*N*) can use a vulnerability on a directly connected and reachable host Y as a stepping-stone for compromising confidential data on host Z reachable only to Y,

$$\forall X, Y, Z \in \mathbf{N} : (X\mathbf{R}Y \wedge Y\mathbf{R}Z) \Rightarrow X\mathbf{R}Z \tag{1}$$

We define a comprehensive list of rules to automate the entire risk analysis process based on the formula as shown in Eq. (2). The terms used in this equation and associated rules are discussed in subsequent sections,

$$\mathbf{I} = \mathbf{L} \times \bar{\mathbf{A}} \times \mathbf{R} \tag{2}$$

Threat relevance with the network (*F*)

There exist several STIX attributes that assist in relating threats with a network. We list here the important attributes (or factors) that we identified as frequently occurring in the data for both threat and network domains. These factors which can be easily extended in the future include vulnerable software, attacker's motivation, location, targeted language, business type, affected assets, affected CIA property, and incident severity. In the following, we discuss how these factors are

Table 1 – Qualifying concepts with notations.

Notations	Definition
F	*Relevance Factors* are the major identified characteristics used to relate threats with the network. Factor i is identified as F_i.
E	Each F_i corresponds to a *Set of Sub-Attributes* that specify its characteristics. E_s symbolizes if E presents in a STIX feed, E_n for the network, and $\bar{E}$ for all E present in STIX.
S	Each F_i has a *Relevance Score*, computed on E. S_i is derived by comparing the similarities between threats target and the victim's network and $\bar{S}_i$ is the maximum of S_i, i.e., 1.
W	Each F_i has its respective W_i that depicts its relevance criticality.
L	*Threat Likelihood* provides the score for the possibility of occurrences of a certain nature of threats (found in STIX) on a network. It is computed based on S_i and W_i.
C	A network comprises quantitative (servers, computers, mobile, etc.) and qualitative (confidentiality, integrity, and availability) assets with their associated *Cost*. C_n denotes the *Affected Quantitative Assets Cost* on the network, targeted by STIX, and $\bar{C}_n$ is for *Total Quantitative Assets Cost* available on the network. C_l represents the *Affected Qualitative Assets Cost* on the network, damaged by threats, and $\bar{C}_l$ is for *Total Qualitative Assets Cost* available on the network.
$\bar{A}$	*Total Loss of Affected Assets* is a collective score that measures the maximum loss with respect to the network assets. $\bar{A}$ is derived from *Quantitative Asset Loss* (A_n) and *Qualitative Asset loss* (A_l). A_n and A_l scores are measured by dividing C_n with $\bar{C}_n$ and C_l with $\bar{C}_l$, respectively.
V	*Scale Value* assigns a criticality score within the range of [0–100] for the affected assets in terms of *Confidentiality*, *Integrity*, and *Availability* with respect to the network. The computation of A_l is done based on V.
R	*Threat Reachability* measures the propagation of threats (attack infiltration) to the vulnerable hosts of network, which are directly or indirectly connected through the Internet. R is achieved by dividing the number of reachable hosts by the total number of available hosts on the network.
I	*Risk Impact* identifies the critical threats that has high impact on the network, derived from L, $\bar{A}$, and R.

populated from the data with the help of deductive inference rules (in *SWRL*) defined in the *STIX-Analyzer*'s ontology. Due to the space limitation, we only mention some significant *SWRL* rules, where we formalize equations to represent the concept of reasoning performed on STIX reports to analyze threats.

Relevance with Software Vulnerability (hasCveRelevance): Exploiting vulnerabilities is a common prerequisite for creating exploits to launch attacks and compromise networks. For example, *CVEs* exploited in recent threats, such as Red October, LUCKYCAT, Naikon and WildNeutron, were employed in multiple kill-chain phases such as for establishing an initial foothold, lateral movement as well as privilege escalation. These are detailed in Table 2. Listing 9 illustrates how a SWRL rule performs vulnerability mapping and measures the *CVE* relevance of reported threats for the network through automated reasoning. The relevance is identified on the basis of vulnerabilities detected from the list of configured services and applications running on the hosts of the network. The rule extracts CVE Id from the *hasCveId* element of STIX. These CVE Ids are then compared with CVE ontology instances to determine the associated list of vulnerable software and version information, as they are defined by *NVD* (NVD, 2015a). Next, the identified vulnerable software with version details are matched with the software installed on the network hosts. If the vulnerabilities identified in STIX exist in the network, the relevance score, i.e., *hasCveRelevance*, for the corresponding CVE element is set.

```
STIX(?stx) ∧
    hasExploitTargets(?stx, ?exp) ∧
    hasVulnerability(?exp, ?vln) ∧
    hasCveId(?vln, ?cid) ∧
CVE(?cve) ∧
    hasCveId(?cve, ?cid2) ∧
    swrlb:containsIgnoreCase(?cid, ?cid2) ∧
    hasVulnerableSoftwareList(?cve, ?vsl) ∧
Network(?ntw) ∧
    hasConnectedHost(?ntw, ?hst) ∧
    hasHostInstalledSoftware(?hst, ?hsl) ∧
    swrlb:containsIgnoreCase(?hsl, ?vsl) ∧
    hasAssociatedRisk(?stx, ?rsk) ∧
    hasSTIXRelevance(?rsk ?rlv)
    → hasVulnerableHost(?ntw, ?hst) ∧
       hasHostVulnerableSoftware(?hst, ?hsl) ∧
       hasCveRelevance(?rlv, 1)
```

Listing 9: Vulnerability Relevance as SWRL Rule

Relevance with Attackers' Motivation (hasMotivationRelevance): Mapping for *hasMotivationRelevance* is designed by relating threat actor's motivations (MITRE, 2014a) provided by STIX to the network. We observe that the motivation of the *Red October* campaign is *Espionage*, the *LUCKYCAT* campaign is *Political*, and the *WildNeutron* APT is *Ego* and *Economic*, as seen in Table 2. Generic mapping rules have been designed to this effect based on domain knowledge of common network security attacks. For instance if the network is part of a financial organization or bank, involved in payment, withdrawals, and other financial transactional services, then these are potential targets of threats where the threat actor's motivation is relevant to *Economic* activities. If the client's network is offering online services which consume high traffic, it can be victimized by an attacker whose motivation is to gain *Publicity*. Attackers with defacement motives will hack and create crafted versions of popular, or publicly accessible websites to launch their propaganda. While this mapping is currently limited to the threat actors' motivation vocabulary (MITRE, 2014a) found in STIX, it is easily extensible.

Relevance with Business Type (hasOrganizationRelevance): The network architecture is strongly dependent on the domain requirements. Network assets, protocols, connectivity and topology will vary significantly depending on whether it is housed inside a home, hotel, library or a power grid substation. It is well known that the cyber kill-chain weaponization phase involves designing malware and exploit toolkits targeting a specific network e.g. malware targeting point of sales (POS) systems will generally target hotel or departmental store networks. Relevance is therefore derived on the basis of a comparison between the targeted industry *OrganizationType* element of the network and the victim's industry type information present in the TTP element of STIX. The *Incident* element of STIX also includes organization names and administrative locations of *ThreatActors* and the *Victim*. This information is also useful to compute relevance. Table 2 presents examples of targeted industries, which include healthcare, banks, military bases, embassies, and law firms.

Table 2 – Observed relevance factors in STIX.

	Relevance Factors F								
	CVE	Motivation	Victim location	Assets	CIA	Language	Organization	Impact	Security compromise
Red October (STIXProject, 2015)	CVE-2008-4250, CVE-2009-3129, CVE-2010-3333, CVE-2012-0158	Espionage	Kazakhstan, Europe, Asia, USA	Desktop, Mobile, Router, Server	Confidentiality Integrity	English	Government, Financial firm Scientific research	High	Yes
LUCKY CAT (FTR-Team, 2012)	CVE-2010-2883, CVE-2010-3333, CVE-2010-3654, CVE-2011-0611	Political	India, Japan	Credit card Banking information, Computers	Confidentiality	Japanese	Military Aerospace, Shipping, Engineering	High	Yes
Naikon (Kaspersky-Team, 2015)	CVE-2012-0158, CVE-2010-3333	Ideological	Myanmar, Vietnam, Malaysia, Philippines, Laos	PCs, Documents, Databases	Confidentiality	Korean	ASEAN governmental agencies, Military, Law enforcement, Embassies	High	Yes
Wild Neutron (GReAT, 2015)	CVE-2012-3213	Ego, Economic	Great Britain, Russia, Switzerland, Germany, Austria	Web application	Availability	English, French	BITSTAMP, Law firms, Bitcoin companies	High	Yes
Givaudan's System (STIX, 2015b)	Unknown	Economic	United States	Server	Confidentiality	English	Unknown	Low	Suspected

Relevance with Target's Location (hasTargetedLocationRelevance): Relevance with the target's location can be judged on the basis of the mailing addresses of both the victim and the attacker. The cyber attack history including sophisticated cyber-attacks and APTs reveals rivalries of a victim. STIX reports contain the threat actor's address field that identifies the location of the attacker, while the targeted victim's address gives the location of the target. Table 2 details victim locations of some recent APTs. *Red October* campaign targets networks in Kazakhstan, Eastern Europe, USA, and Central Asia, while *WildNeutron* attacks the locations in Great Britain (UK), France, Russia, Switzerland, Germany, and Austria and finally the *Naikon* APT targets South-eastern Asia and areas around the South China Sea. In Listing 10, relevance is derived by analyzing the target's physical location. For identity characterization of the incident victim and the threat actor, STIX uses OASIS Customer Information Quality (CIQ) (Stixproject, 2015). The CIQ specification field contains information regarding the address, locality, country, and administrative area and this information is used in the rule to derive location relevance with the network location identified using geo-location through *hasGeolocationCountry* elements.

```
STIX(?stx) ∧
    hasIncident(?stx, ?inc) ∧
    hasVictim(?inc, ?vct) ∧
    hasAddress(?vct, ?adr) ∧
    hasCountry(?adr, ?cnt) ∧
Network(?ntw) ∧
    hasGeoLocationCountry(?ntw, ?glc) ∧
    swrlb:stringEqualIgnoreCase(?cnt, ?glc) ∧
    hasAssociatedRisk(?stx, ?rsk) ∧
    hasSTIXRelevance(?rsk, ?rlv)
    → hasTargetedLocationRelevance(?rlv, 1)
```

Listing 10: Location Relevance as SWRL Rule

Relevance with Affected Assets (hasAssetsRelevance): Relevance with affected assets is identified either by the incident type as classified in the STIX report or by finding a match between the targeted asset specified in STIX with the asset configuration of the network. Affected assets include PCs, mobile phones, databases, servers, credentials, and records etc. provided in STIX incident *AffectedAssetType* vocabulary (MITRE, 2013e). Table 2 highlights the compromised assets of *Red October*, *LUCKYCAT*, *WildNeutron*,*Givaudan's System* and *Naikon*. We define a rule that compares the network's assets with the instances of STIX reports in order to estimate the damage caused by a particular threat in Listing 11.

Relevance with Compromised CIA Property (hasCiaRelevance): The relevance of a compromised CIA property with the network depends on the organization's preference for a particular CIA property. It is derived from the *hasPropertyAffected* of STIX (owl) which provides the CIA information. Table 2 shows that the *Red October* APT compromises the *Confidentiality* and *Integrity* of the network. *LUCKYCAT*, *Naikon* and *Givaudan's System* targets the *Confidentiality* and *WildNeutron* affects the *Availability* of the network. We define a rule to identify the threats that affect or target the specific CIA property: confidentiality, integrity, or availability of network assets.

```
STIX(?stx) ∧
    hasIncident(?stx, ?inc) ∧
    hasAffectedAssets(?inc, ?ast) ∧
Network(?ntw) ∧
    hasNetworkAssets(?ntw, ?nst) ∧
    swrlb:containsIgnoreCase(?nst, ?ast) ∧
    hasAssociatedRisk(?stx, ?rsk) ∧
    hasSTIXRelevance(?rsk, ?rlv)
    → hasAssetsRelevance(?rlv, 1)
```

Listing 11: Assets Relevance as SWRL Rule

Relevance with Incident Severity (hasImpactRelevance): The severity factor is described by the *hasImpactAssessmentType* property of STIX (owl). Impact assessment effect vocabulary (MITRE, 2014b) includes "financial loss" and "loss of competitive advantage – military" etc. defined by a *hasConfidence* value that reflects the level of threat impact and range in terms of high, medium, and low. We define a rule for deriving the severity in Listing 12. The rule filters STIX reports where the confidence value is *High*.

```
STIX(?stx) ∧
      hasIncident(?stx, ?inc) ∧
      hasImpactAssessmentType(?inc, ?iat) ∧
      hasConfidence(?inc, ?cnf) ∧
      swrlb:stringEqualIgnoreCase(?cnf,"High") ∧
Network(?ntw) ∧
      hasAssociatedRisk(?stx, ?rsk) ∧
      hasSTIXRelevance(?rsk, ?rlv) ∧
      → hasImpactRelevance(?rlv, 1)
```

Listing 12: Severity Relevance as SWRL Rule

Relevance with Language (hasTargetedLanguageRelevance): A majority of attacks target the network and people on the basis of their nationality and language. We define a rule in Listing 13 that detects the language provided in the imported network's configurations through the regional and languages customization features of installed software (e.g., the language of the operating system). This is compared with the targeted victim's language in the STIX report. The targeted languages shown in Table 2 are English, French, Japanese and Korean.

Relevance with Security Compromised Element (hasSecurityCompromisedRelevance): The "security compromised" element indicates a successful security breach incident as apposed to a failed attempt. To this effect, ontology rules automatically reason that STIX (owl) reports containing a value of "Yes" for the *hasSecurityCompromise* property have higher consideration than others.

These nine major relevance computing factors are denoted using F_i where $(0 \le i \le 9)$. It is common to have only a few of these factors being observed together at any one time in any one particular STIX report. If any of these elements is missing from a report, the rule finds the relevance using other identified elements to map threats to the network.

```
STIX(?stx) ∧
    hasIncident(?stx, ?inc) ∧
    hasVictim(?inc, ?vct) ∧
    hasIdentity(?vct, ?id) ∧
    hasLanguage(?id, ?lng) ∧
Network(?ntw) ∧
    hasConnectedHost(?ntw, ?hst) ∧
    hasSoftwareInstallationLanguage
    (?hst, ?sil) ∧
    swrlb:stringEqualIgnoreCase(?lng, ?sil) ∧
    hasAssociatedRisk(?stx, ?rsk) ∧
    hasSTIXRelevance(?rsk, ?rlv) ∧
    → hasTargetedLanguageRelevance(?rlv, 1)
```

Listing 13: Language Relevance as SWRL Rule

Threat likelihood (*L*)

Threat Likelihood (*L*) is measured using a score that represents the possibility of occurrences of certain nature of threats on the network. The calculation of *L* is based on the relevance scores (S_i)s.

From nine major identified factors F_i, the presence of each relevance factor is counted as a single unit. If the sum of the combined relevance score is greater than or equal to one $(\sum_i S_i \ge 1)$ i.e. the lowest possible relevance, only then the respective STIX is considered as relevant. The maximum sum of relevance score could be nine $(\sum_i S_i \le 9)$. Each factor F_i contributes with a set of sub-attributes *E* to the level of similarity between the targeted threat (according to the STIX report) and the victim's network. The number of common attributes between STIX and the network instances ($E_s \cap E_n$) is divided by the number of total available attributes ($|\bar{E}|$) to measure S_i for F_i. Equation (3) is used to calculate the relevance score. The maximum value of S_i is 1,

$$S_i = \frac{E_s \cap E_n}{|\bar{E}|} \quad (3)$$

where
E_s is the set of relevance elements found in STIX
E_n is the set of relevance elements received in the network
$\bar{E}$ is the set of all available relevance elements

In order to understand the concept, consider the rule defined in Listing 9. This allows the detection of vulnerabilities in the network by finding a match between vulnerable software as defined in STIX reports and the software programs or services installed on the hosts. To compute the relevance score S_i for the vulnerability factor, the number of common vulnerable software is divided by the total number of software vulnerabilities present in STIX reports. Similarly, Listing 10 compares the locations of the targeted STIX report E_s and the victim's network location elements E_n. The CIQ element of STIX is used to derive the location relevance. The number of matched ($E_s \cap E_n$) location elements is divided by the total available ($\bar{E}$) location identification elements, and the result is stored as S_i where *i* depicts the location.

Weights (W_i) are assigned to relevance factors (F_i) in order to realize the criticality and impact of F_i with respect to the network. These weights are configurable by the network administrator. S_i is derived from Eq. (3) and W_i is assigned to each F_i. Equation (4) calculates *L* based on all F_is, i.e., S_is and their corresponding W_is. *L* is calculated by summing the products of S_is with their corresponding W_is and then dividing the sum by the summation of the product of the maximum relevance score ($\bar{S}_i$) and W_i of the respective relevance factor F_i,

$$L = \max_{0 \le S_i \le 1} \frac{\sum_{i=0}^{N} S_i \times W_i}{\sum_{i=0}^{N} \bar{S}_i \times W_i} \quad (4)$$

where
N is number of relevance factors *F*
S_i is received relevance score for F_i
$\bar{S}_i$ is maximum relevance score for F_i
W_i is assigned weight for F_i

Total loss of affected assets ($\bar{A}$)

The total loss of the affected assets $\bar{A}$ is calculated by analyzing both quantitative and qualitative asset loss in the network. The modeled network assets that can be analyzed quantitatively are personal computers, servers, cell phones, routers, and switches. The formulated rules are used to derive the number of affected assets targeted by threat sources on the victim's network (as discussed in Listing 11). In Eq. (5), *Quantitative Asset Loss* A_n is derived by dividing *Affected Quantitative Assets Cost* C_n from *Total Quantitative Assets Cost* $\bar{C}_n$. Therefore, we define qualitative asset levels with respect to a *scale*

value V that scales the quantitative assets to vary between the ranges of 0 and 100 based on their criticality in terms of the CIA requirements. The product of V and C_n provides the *Affected Qualitative Assets Cost*, C_l. We divide C_l by the *Total Qualitative Assets Cost* $\bar{C}_l$ to produce A_l. The imported network instances in the framework carry V for each host. If V is undefined on the network instance, the framework assumes the network asset as non-critical and uses only the A_n as the $\bar{A}$. A_l measures the worth of a critical resource present in the network with respect to CIA and is crucial in deriving the impact (*I*). The framework measures A_n and A_l through reasoning. *Total Loss Of Affected Assets* ($\bar{A}$) is calculated by dividing the sum of A_n and A_l by 2 (the maximum score of A_n and A_l). The corresponding *swrl* rule that allows this automatic derivation is shown in Listing 14,

$$A_n = \frac{C_n}{\bar{C}_n}, \qquad C_l = C_n \times V, \qquad A_l = \frac{C_l}{\bar{C}_l}, \qquad \bar{A} = \frac{A_n + A_l}{2} \tag{5}$$

where

A_n is Quantitative Asset Loss
C_n is Affected Quantitative Assets Cost
$\bar{C}_n$ is Total Quantitative Assets Cost
C_l is Affected Qualitative Assets Cost
V is Scale Value
$\bar{C}_l$ is Total Qualitative Assets Cost
A_l is Quantitative Asset Loss
$\bar{A}$ is Total Loss Of Affected Assets

Threat reachability (*R*)

Threat Reachability (often termed as exposure) measures the accessibility or infiltration of threats to network vulnerable hosts, which are directly and indirectly connected to the Internet. Threat reachability denoted as *R* determines threat impact *I* on the network architecture by identifying the number of assets exploited and exposed to a particular threat. All vulnerable hosts may not be exploitable in the network due to restricted reachability. For example, appropriate deployment of firewalls can isolate threats and attack escalation using access control rules to restrict access from the Internet to the hosts in the network. We assume in the *STIX-Analyzer* ontology that if a firewall is activated for the vulnerable hosts, then those hosts are not reachable. By performing the reachability analysis, we can determine the threat propagation and, thereby, the security state for particular hosts of the network by determining the network links, traffic flow, and activated controls.

```
STIX(?stx) ∧
    hasAssociatedRisk(?stx, ?rsk) ∧
    hasQuantitativeAssetsLoss(?rsk, ?qnl) ∧
    hasQualitativeAssetsLoss(?rsk, ?qll ) ∧
    swrlb:add(?rzlt, ?qnl, ?qll) ∧
        swrlb:divide(?rzlt2, ?rzlt, 2) ∧
    → hasTotalLoss(?rsk, ?rzlt2)
```

Listing 14: Total Loss as SWRL Rule

To map the concept formally, we have defined rules for reachability that enumerate the possible paths from the threat source to hosts or from affected hosts to other (vulnerable) hosts. The rules are designed as a collection of several steps: (a) identifying vulnerable hosts, (b) detecting reachable routers connected with the vulnerable hosts, and (c) determining the subnets or hosts connected with these reachable routers. Vulnerable hosts (PCs and servers) in the network are identified in Listing 9. The identified vulnerable hosts are used in Listing 15 to detect the exposed and reachable routers linked with the directly connected vulnerable host. Further reachability rules defined in *STIX-Analyzer* find the hosts connected with reachable routers to identify the exposed subnets. The security state is measured from the reachability score (*R*), which is the result of dividing the number of vulnerable hosts with the total number of hosts present in the network. The reachability score lies between 0 and 1, representing lowest and highest values respectively. For instance, if no control is enabled for a network host that contains all STIX targeted vulnerabilities then it is scored as one (i.e., the threat reachability is high).

```
Network (?ntw) ∧
    hasRouters(?ntw, ?rtr) ∧
    hasConnectedHost(?rtr, ?hst) ∧
    hasVulnerableHost(?ntw, ?vhst) ∧
    swrlb:equal(?hst, ?vhst)
    → hasReachableRouters(?ntw, ?rtr)
```

Listing 15: Routers Reachability as SWRL Rule

Finally, in order to calculate the impact, the scores corresponding to the threat likelihood, affected asset loss, and reachability are multiplied together, as shown in Eq. (2). To elaborate the concept further, in the following section we discuss the campaign of Red October as a case study on our sampled network.

3.5. *Case study: Red October STIX impact on network*

The Red October campaign (GReAT, 2013) encompasses a series of cyber attacks targeting government and financial organizations. It exploits vulnerabilities present in Microsoft Office Word and Excel to gain a foothold in the network. We consider the Red October STIX report (STIXProject, 2015) as a case study to derive its impact on a sample network based on the proposed *STIX-Analyzer* framework. The topology of the sample network is shown in Fig. 4.

According to the parsed Red October STIX report, a STIX instance is generated. The factors in this particular instance: *Exploit-Targets* include three major vulnerabilities associated with MS Word and Excel. The *CveId*(s) of these vulnerabilities are: CVE-2012-0158 (vulnerable products: MS Office 2007, 2010), CVE-2009-3129 (vulnerable products: MS Excel 2003, 2007, 2010 Gold), and CVE-2010-3333 (vulnerable products: MS Office 2003, 2007, 2010). MS Office Word and Excel 2007 are installed on three hosts namely: PC-X, PC-Y and PC-Z. As both MS Word and MS Excel vulnerabilities are present in the network to trigger the exploit, S_i for each vulnerability is 1. The report also specifies 5 types of assets: PCs, switches, servers, routers, and cell phones that are affected by this attack. Four types of these assets (PCs, switches, servers, and routers) are present in the sample network. Therefore, the relevance score (S_i) where *i* is for assets is $\frac{4}{5}$, i.e., 0.8. The motivation of the Red October attack, as specified in the STIX feed, is *Finance* and it targets US government agencies and financial organizations. Since the sample network's organization type is "Health-Sector", S_is for *Motivation*, and *Business Type* relevance factors are 0. The targeted *Country* element is "USA" and targeted *Language* is "English", which are compared to *hasGeolocationCountry*, and *InstalledSoftwareLanguage* elements present in the network instance, respectively. The

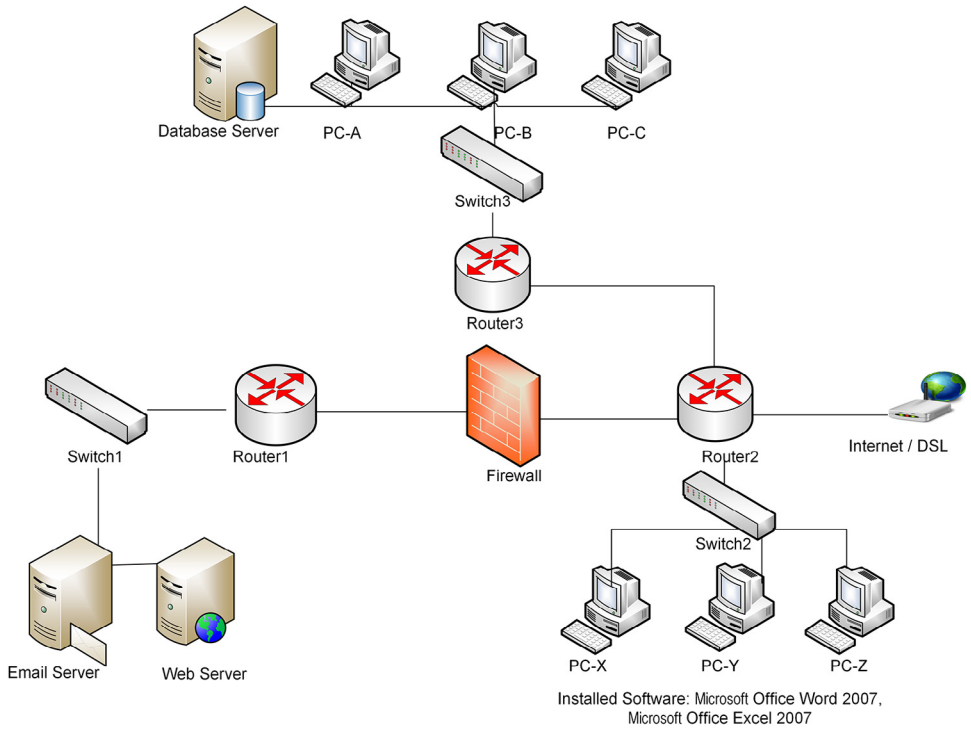


Fig. 4 – Network example.

matched S_is for *Country* and *Language* are set to 1, while the *LocationRelevance* factor has sub attributes E_n (address, locality, administrative area) that are missing in STIX (E_s), resulting in $S_i = 0.25$ for the location. The sample health-sector network contains confidential data, where the availability as well as the integrity of the records are important (as modeled in the *hasCiaPreference* element of network). The *hasCiaPreference* is compared with *hasAssetProperty* of the Red October's ontology instance. The derived score of S_i for *CIA* is 1, which specifies that the preferred CIA properties in the network and targeted CIA properties of STIX are the same. The relevance score computation is performed by the rules defined in *STIX-Analyzer* considering the ontology instances of the Red October STIX report and that of the network, as shown in Table 3.

While the weights are customizable based on the security analyst's judgement, we define them for our prototype according to the following criteria. The maximum weight (W_i) is assigned to *hasCveRelevance* (F_i) because the attack can only trigger if the STIX targeted vulnerability is present in the network and early identification of cve relevance of STIX is crucial. The second highest weight is assigned to the affected assets relevance (*hasAssetsRelevance*) and the CIA relevance (*hasCiaRelevance*) which measures the targeted and network associated quantitative and qualitative assets, respectively. A STIX report does not merit serious consideration, if the affected asset (attacher's goal) is not present in the network. The third highest weight is assigned to country relevance element (*hasTargetedLocationRelevance*) as modern-day APTs and sophisticated attacks are mostly tailored toward specific regions or people of the targeted or neighboring locality as apposed to aiming for global domination. The motivation relevance is placed at the fourth position which is crucial to identify the STIX reports that match the network's intent. The rest of the four factors are of secondary important and thus their weights are assigned to 1. Table 4 depicts the weights (W_i)s of the relevance factors (F_i)s.

The computed relevance scores (S_i)s are shown in Table 3. The associated weights (W_i)s are the same as presented in Table 4. The maximum relevance score, denoted as $\bar{S}$, is 1 for all relevance computing factors (F_i)s, as explained in Section 3.4. Based on these scores, the threat likelihood score (L) is calculated using the rule as specified in Eq. (4). The calculation procedure of L in *STIX-Analyzer* is rendered in Table 5. The

Table 3 – Network STIX relevance.

	Red October STIX elements (E_s)	Network elements (E_n)	Relevance score (S_i)
hasCveRelevance	CVE-2012-0158, CVE-2010-3333, CVE-2009-3129	CVE-2009-3129, CVE-2010-3333, CVE-2012-0158	1
hasAssetsRelevance	Servers, routers, switches, PCs and mobile phones	Servers, routers, switches and PCs	0.8
hasTargetedLocation Relevance	USA	USA	0.25
hasTargetedLanguage Relevance	English	English	1
hasCIA-Relevance	Confidentiality, integrity, availability	Confidentiality, integrity, availability	1

Table 4 – Relevance factors and associated weights.

F_i	W_i
hasCveRelevance	5
hasAssetsRelevance	4
hasCiaRelevance	4
hasTargetedLocationRelevance	3
hasMotivationRelevance	2
hasOrganizationRelevance	1
hasImpactRelevance	1
hasTargetedLanguageRelevance	1
hasSecurityCompromiseRelevance	1

likelihood score for the Red October attack on the sample network instance is 0.63.

The identified quantitative network affected assets, such as PCs, servers, switches, and routers, their costs, and the ultimate quantitative cost (C_n) calculation are presented in Table 6. Likewise the CIA critical qualitative network affected assets, their values, and the corresponding qualitative cost (C_l) calculation are shown in Table 7. In this case study, as shown in Table 7, servers require confidentiality, availability, and integrity guarantees, while PCs and routers require mainly availability. The total loss of affected assets $\bar{A}$ for the network is measured using Eq. (5), and the respective calculations are shown in Table 8.

Table 5 – Threat likelihood (*L*).

F_i	$S_i \times W_i$	$\bar{S} \times W_i$
hasCveRelevance	5	5
hasAssetsRelevance	3.2	4
hasCiaRelevance	4	4
hasTargetedLocationRelevance	0.75	3
hasMotivationRelevance	0	2
hasOrganizationRelevance	0	1
hasImpactRelevance	0	1
hasTargetedLanguageRelevance	1	1
hasSecurityCompromiseRelevance	0	1
SUM	13.95	22
L		13.39/22 = 0.63

Table 6 – Quantitative asset cost (C_n).

	Number of affected assets	Cost per asset	Cost per asset × Number of assets
PCs	6	$ 1090	$1090 × 6 = 6540
Switch	3	$ 1590	$1590 × 3 = 4770
Server	3	$ 4593	$4593 × 3 = 13,779
Routers	3	$ 1998	$1998 × 3 = 5994
C_n			USD $31,083

Table 8 – Total loss of affected assets ($\bar{A}$).

C_n	$\bar{C}_n$	A_n	C_l	$\bar{C}_l$	A_l	$\bar{A}_l$
31,083	35,211	0.88	4,860,840	4,860,840	1	0.94

Threats can propagate to various hosts of the network from the Internet. In the sample network illustrated in Fig. 4, various paths are identified from the Internet to network hosts by executing the reachability rule. The attack will infiltrate the network by exploiting the vulnerabilities of hosts directly and indirectly connected with the Internet through Router 2. An exploit from Router 2 will penetrate the network by targeting the hosts connected with Router 3, which is indirectly connected to the Internet through Router 2 and targets the attached database server. As the firewall is activated for Router 1, it blocks the threat accessibility to the connected email server and web server from the Internet and vulnerable hosts of Router 2. As all hosts except the hosts connected with Router 1 are affected, the reachability score (R) is $\frac{11}{17} = 0.64$. The impact I according to Eq. (2) is derived as $0.63 \times 0.94 \times 0.64 = 0.37$.

4. Threat attribution

The previous sections demonstrated how the mapping of threats to a network allows for a meaningful analytsis such as risk and impact assessment. Another useful analytics, that we term as *threat attribution*, maps threats with other similar threats in order to aid sense-making and decision making. In our experiments we populated our ontology from voluminous threat repositories (FS-ISAC, 2015; STIXProject/schemas-test, 2015; TAXII, 2015). The proposed ontology model allows insights that are otherwise not possible when examining a single report at a time. Threat attribution allows for the creation of threat profiles to proactively enforce network security controls based on commonly observed motives, goals, attack patterns, tools, and methods.

Among many, we found that the properties of the *Observables* STIX construct are very useful for threat attribution in terms of discerning (i) threat frequency and (ii) threat actor's profile. While a number of attributes can be useful for the threat actors attribution, we identified fourteen major elements within STIX for our work that included a threat actor's *Name*, *CampaignTitle*, *AddressObject*, *URIObject*, *IP*, *Country*, *AdministrativeArea*, *OrganizationType*, *Language*, *Threat Actors Type Motivation*, *EmailMessageObj*, *NetworkConnectionObj* and *HttpSessionObj*. An

Table 7 – Qualitative Asset Cost (C_l).

	Critical Assets	Critical assets cost × Number of assets	V [0–100] × Critical assets cost × Number of assets
Confidentiality	Server	$13,779	100 × 13,779 = 1,377,900
Availability	PCs, server, routers	$6540 + $13,779 + $5994 = $26,313	80 × 26,313 = 2,105,040
Integrity	Server	$13,779	13,779 × 100 = 1,377,900
C_l			USD $4,860,840

evaluation of threat actor's attribution is shown in Fig. 7 and discussed in Section 5.1.3. The *CybOX* sub-class of *Observables* provides *hasAddressObj* for determining the attacker's IP address and domain name, *hasURIObject* for malicious URLs, *hasEmaiMessageObj* for phishing emails with attachments and file extension details, *hasNetworkConnectionObj* for protocol information, and *hasSocketAddressObj* for socket addresses including the IP information. Next, we discuss how these properties help in identifying common attack patterns and associating threat actors.

4.1. *Threat frequency analysis*

Threat frequency analysis correlates multiple STIX instances based on frequency of occurrences of a particular property, such as the threat actor domain. A malicious domain that frequently co-hosts multiple malwares or acts as command and control for a variety of attacks can be prioritized for blocking even if it is not currently relevant to the network in question (may become relevant in the future). In Listing 16, the *hasAddressObj* property of *CybOX* is used to determine the occurrence of the threat actor's domain of STIX instance (*STIX_Inst*) with other STIX available instances in the knowledgebase. At every occurrence of the threat actor's domain in STIX, the frequency count is incremented by one and the resulting value is stored in the *hasAnalyzedFrequency* property. Threat domains are blocked in the network via access controls, i.e., by enabling firewalls to block malicious URIs, IPs and domain names. We define a rule to formalize a traffic blocking mechanism by adding STIX identified malicious domains to *hasDenyList* or *hasBlockList* of the firewall policy.

```
STIX(STIX_Inst) ∧
    hasObservables(STIX_Inst, ?obs) ∧
    hasCybOX(?obs, ?cbx) ∧
    hasAddressObj(?cbx, ?abj) ∧
STIX(?stx) ∧
    hasObservables(?stx, ?obs2) ∧
    hasCybOX(?obs2, ?cbx2) ∧
    hasAddressObj(?cbx2, ?abj2) ∧
    swrlb:equal(?abj, ?abj2) ∧
    hasAnalyzedFrequency(STIX_Inst, ?frq) ∧
    swrlb:add(?newfrq, ?frq, 1)
    → hasAnalyzedFrequency(STIX_Inst, ?newfrq)
```

Listing 16: Frequency Analysis as SWRL Rule

4.2. *Threat actor profiling*

In our experiments, we observed that attacks within a single campaign usually follow specific attack patterns and share the same goals and motivations. Alerts are generated by comparing the known indicators, malicious email information, malware hashes, and signatures present in the STIX indicators. An example of such a campaign is *Lizard Squad*. This campaign is observed to launch mostly DDOS attacks. If this name is detected as *PartyName* of *ThreatActors* in frequent and high impact threats for the network, an alert will be generated. Similarly *Motivation* is analyzed to identify the correlation between the attacker's intent and corresponding actions. As an example of this analysis, we found multiple STIX reports where the attacker's motivation was *hacktivism*. Attacks described in these reports tried to damage the reputation of the organization by propagating an agenda or launching a political movement through defacement attacks. Similarly, we found *ideological* or *politically* motivated attacks like Poison-Ivy (STIX, 2015a) and Shadyrat (Sherstobitoff and Liba, 2015), which can create high impact with long durations. A common threat victim country was USA and targeted language was English as in the case of IXESHE (Sancho et al., 2012), Red October (GReAT, 2013), and Shadyrat APTs. We also observed that many of the attacks, such as Wild Neutron (GReAT, 2015), POS (Point-of-Sale) Malware (Yaneza, 2015), and SOE (Sony Online Entertainment) (InfoSec, 2011) were launched with economic and financial motivation.

5. Evaluation

We evaluate *STIX-Analyzer* to measure the effectiveness and efficiency of the implementation. The effectiveness is evaluated in terms of clarity, consistency of the ontology structural design, accuracy of the reasoning and by feature comparison with competing systems. The efficiency on the other hand reports how the system scales in terms of performance when processing increasing volumes of threat reports and network descriptions. Performance is computed in terms of inference time, processor utilization, and memory reservation.

Since the ultimate end-goal is to aid the cyber-security analyst in conducting network analytics, we also conducted a user study. The survey results determine how *STIX-Analyzer* fares in terms of usability, ease of configuration, and usefulness of results.

The evaluation process confirms the usefulness of the proposed framework along with its compliance with existing CTI frameworks (STIX, CybOX, and CVE) and various network architectures.

5.1. *Effectiveness*

The basic aim of following an ontological approach in the design of *STIX-Analyzer* is to express the complex knowledge of the cyber threat intelligence domain in a way that is computationally traceable and machine processable. A key advantage gained over non-ontological systems is that it provides inference ability and the necessary constructs that enable software agents to reason over the knowledge base. In order to provide accurate and meaningful results, it is important to verify that the design of the ontology model can perform the desired functional requirements correctly in the real world. In the following we describe how our evaluation verifies whether the defined concepts are clear and consistent with other concepts and that the knowledge can be accurately inferred from the model correctly.

In order to evaluate the ontology design structure, we utilize evaluation plug-ins (Tantsis and Kehagias, 2013) supported by *Protege* (Protege, 2014). The evaluation metrics provide details regarding the hierarchy of classes, the minimum and maximum numbers of parent, sibling and child classes defined in the framework, and the number of object and data properties. The number of domain specific properties is 368. The total number of annotations and imposed restrictions of various types including existential, minimum, maximum, hasValue, and universal are also given. The counts of different OWL entities shown in Table 9 are fixed except the individual count and the

Table 9 – Ontology structure.

Owl entities	Count
Classes count	95
Max parents classes	5
Max siblings classes	15
Object property count	108
Data type property count	260
Individual count	2,500
(Total) Number of restrictions	12,263

total number of restrictions. The number of restrictions and inferred properties increases with the use of properties and values present in the newly imported instances on which these restrictions are imposed.

5.1.1. *Clarity*

Our proposed *STIX-Analyzer* framework utilizes the knowledge of multiple components and the naming conventions used for properties and concept labeling are self-descriptive and easy to follow. This helps in grasping and analyzing various concepts used in the developed framework. OntoClean (OntologWiki: Onto Clean, 2004) and Ontology Evaluation (Tantsis and Kehagias, 2013) tabs are used to improve and evaluate the clarity of the designed ontology by refining and validating the general owl concepts to deliver the intended meaning.

5.1.2. *Consistency*

STIX-Analyzer is logically consistent with respect to all the concepts, classes, relationships between properties and instances. We utilize the Pellet reasoner (Complexible/pellet, 2013) to perform the consistency check by evaluating the relationships between classes, subclasses, individuals, objects, data, functional properties, and restrictions. The reasoner detects and identifies incomplete and conflicting properties in the list. Using Pellet, no flaw or inconsistency is found in the proposed model (as depicted in Fig. 5). The reasoner takes only a few seconds to perform the consistency check.

5.1.3. *Reasoning accuracy*

In order to evaluate the reasoning accuracy, various threat reports in both unstructured (APT reports in pdf format) and structured (STIX xml files) formats are parsed and imported as STIX ontology instances and analyzed by the proposed framework. Based on these STIX instances, reasoning of *Impact* and *Threat Actors Attribution* is performed. After analyzing multiple STIX feeds, we find that the advantage of employing different relevance computing factors (F) allows the rule engine to identify the relevance and threat attribution to a high degree of accuracy through the ontology reasoning on defined rules.

Relevance Identification:

Various structured and unstructured threat reports of different sizes are analyzed to identify the quality of different relevance attributes present in the reported threats that are required to perform reasoning. Thirty recent advanced persistent threat reports are considered to analyze the quality of relevance attributes, particularly *CVE*, *Motivation*, *Location*, *Assets*, *CIA*, *Language*, *Organization*, *Impact*, and *Security Compromise* (as discussed in Section 3.4). Fig. 6 represents the quality of STIX feeds with reference to the presence of relevance factors found in reported threats. We observe that necessary attributes required for identifying all (nine) relevance factors are present in APT reports of Red October (STIXProject, 2015), LUCKYCAT (FTR-Team, 2012), Naikon (Kaspersky-Team, 2015), APT1 (STIX, 2015c), and Poison Ivy (STIX, 2015a). However most of the STIX feeds and threat reports such as Givaudan's System (STIX, 2015b) are found to be sparse and incomplete. The least number of *Relevance Factors* were found in the STIX feed related to the FBI Investigation where multiple banks were compromised (Goldstein et al., 2014) and the resultant relevance score for this particular STIX feed is 3. We configure *STIX-Analyzer* to mark reports as irrelevant, if the relevance attributes corresponding to network design (S_i) are found to be less than a threshold of 1.

Threat Actors Attribution:

To assess the quality of threat attribution, various constructs for threat actor identification are analyzed on imported

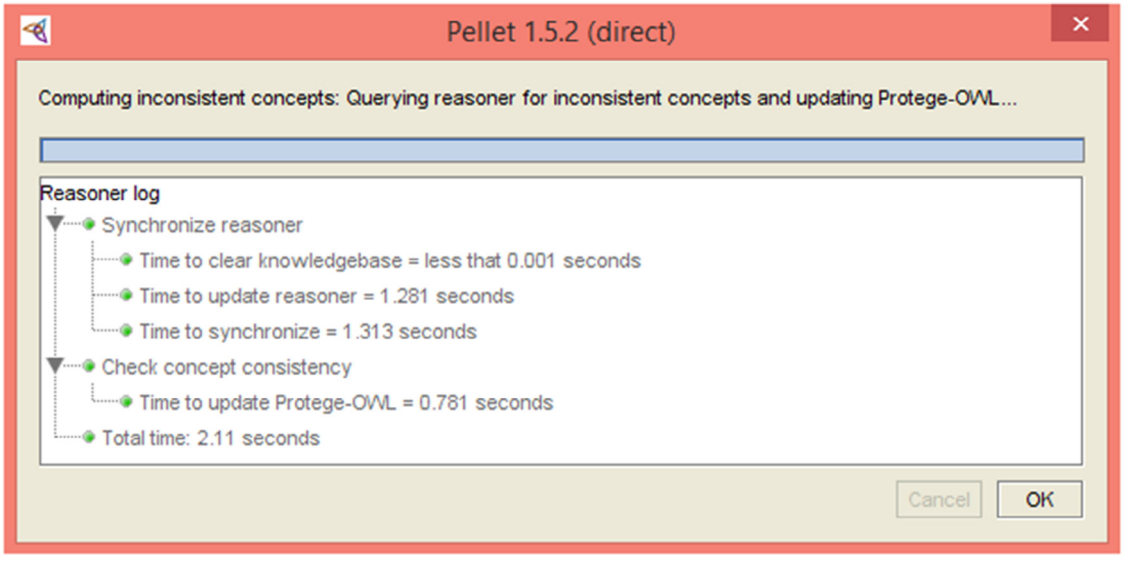


Fig. 5 – Consistency evaluation.

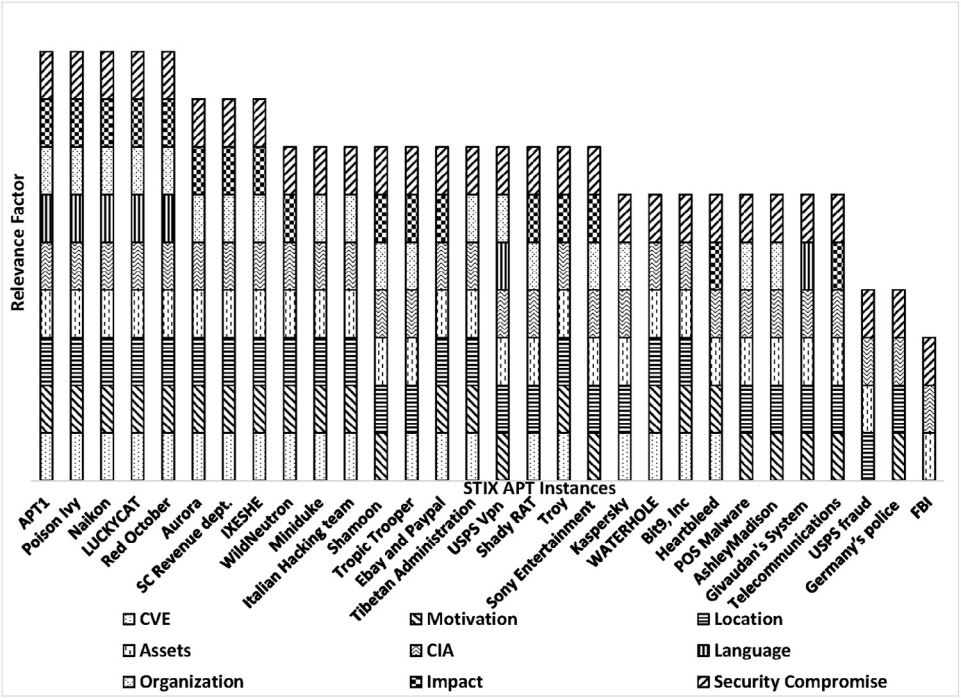


Fig. 6 – Relevance factors (F) found in STIX.

instances of STIX reports. After analyzing multiple STIX reports in our experiments, we find that many of the necessary elements are only sparsely available. Fig. 7 shows the observed results of thirty STIX instances with their associated attribution elements (e.g., threat actor's *Name*, *Country*, *Motivation*, etc.) that are used by the framework. We observe that the CIQ element that represents the threat actor's identity is present in most of the STIX feeds. Thirteen out of fourteen attributes for threat actors attribution are found in *LUCKYCAT* campaign, while only the threat actor's *Motivation* is detected in reports regarding POS Malware. This is because many organizations sometimes prefer not to openly publish attacker names or victim countries.

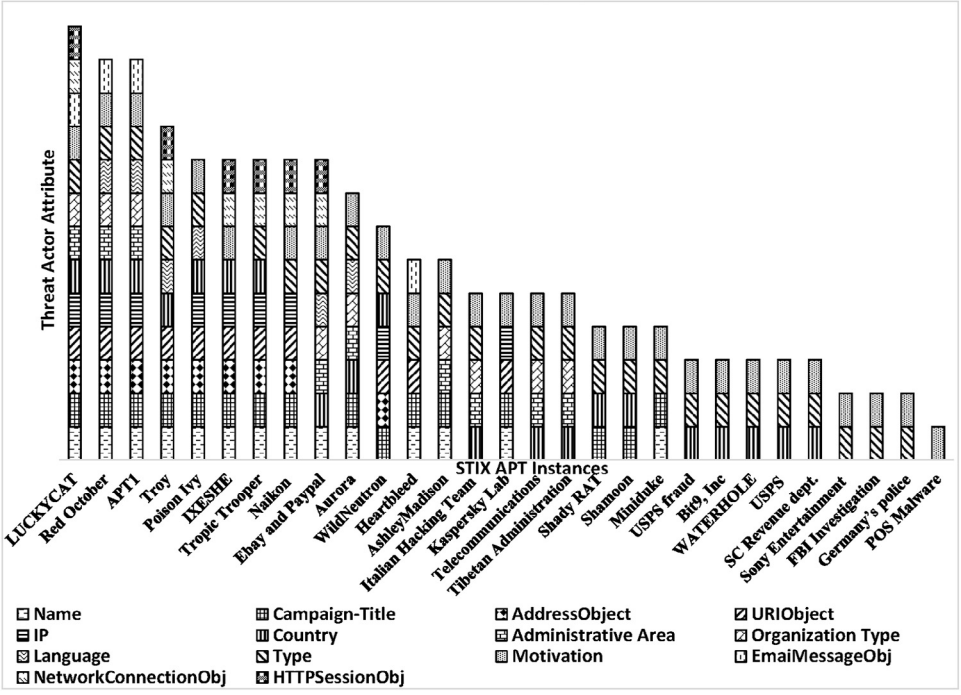


Fig. 7 – Threat actor's attributes found in STIX.

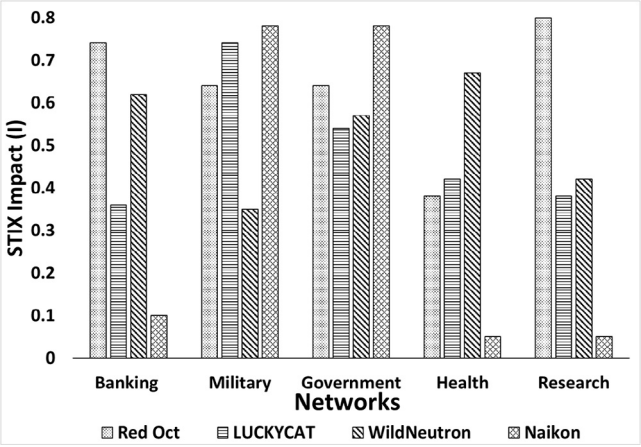


Fig. 8 – Relevance factors (F) found in networks.

Network Impact Analysis:

We have evaluated the impact *I* of multiple real time STIX on various networks. The repository of STIX and network instances are placed online (Qamar and Anwar, 2016), a small sample of which is shown in Fig. 8. The impact of STIX reports, namely, Red October (STIXProject, 2015), Luckycat (FTR-Team, 2012), Wild Neutron (GReAT, 2015), and Naikon (Kaspersky-Team, 2015), is derived on networks of various types of roughly the same size, including banks, military, government, health, and research institutions. The results show that the highest impact of Red October is seen on scientific research organizations because the targeted organization, assets, exploited pdf and office vulnerabilities were available in the configured network. Similarly, significant impact of Luckycat is observed on military networks as most of the identified relevance attributes given in Table 2 matched. The impact of the Wild Neutron campaign is high on health and banking sectors and the Naikon attack equally targets networks belonging to military and government organizations.

5.1.4. *Comparative analysis*

Our proposed *STIX-Analyzer* framework is a novel ontology-based solution that performs automated analysis, such as network relevance identification, impact assessment, and threat attribution, using structured (STIX) and unstructured threat feeds for given network configurations. Due to its novelty, we were unable to perform a direct comparison with other existing systems, as the current state-of-the-art frameworks either solely focus on analyzing cyber-threats or directly study network vulnerabilities without consideration of the prevalent cyber-threat landscape. Nevertheless, a detailed comparative analysis (Table 10) of *STIX-Analyzer*'s features has been conducted with the most closely related frameworks that we could find available as open-source software. *STIX-VIZ* (Mitre, 2013a) imports threat reports in the STIX format and helps visualize the high level constructs as trees and graphs. Its abilities are limited to basic visualization of small files. *Soltra Edge* (DTCC and FS-ISAC, 2015) acts as a threat sharing repository that supports STIX based threat reports. It also supports automatic blocking of threat indicators by creating Snort rules (Soltra Solutions, 2015). However, it does not consider the network configuration. Another comparable system is *Spiderfoot* (SpiderFoot, 2012a) which performs semi-automated penetration testing and foot printing of a supplied target. It supports integration of multiple data sources (SpiderFoot, 2012b) to identify malicious IPs, traffic and vulnerabilities. *Spiderfoot*, however, does not use threat reports to perform analytics.

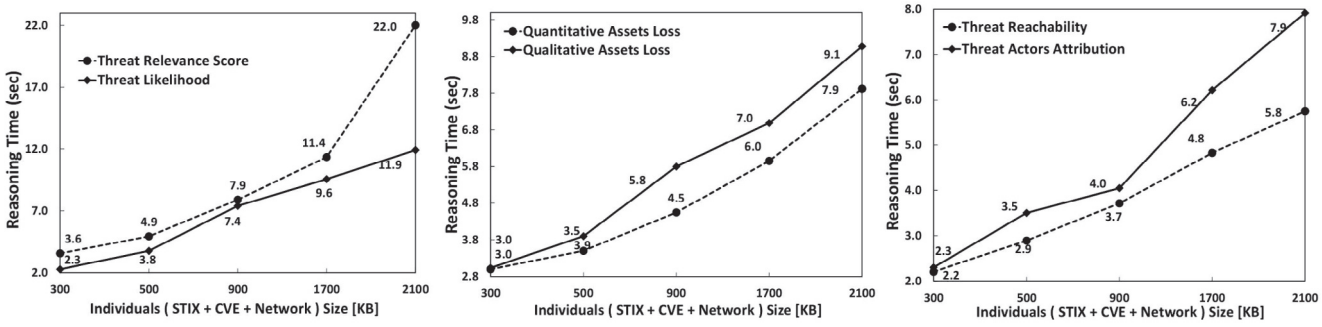
5.2. *Efficiency*

The efficiency of *STIX-Analyzer* is evaluated by importing a set of recent threat reports as instances in xml format as discussed in Section 3.2 and analyzing the performance of the automated reasoning process on a set of imported network schematics. The size of the instances (in kB) is used to evaluate the performance based on the attributes and elements defined. A 100 kB CVE repository is considered in the performance evaluation experiments. Threat reports of varying sizes from 10 kB to 250 kB are imported as STIX xml files, which are made available by open threat repositories directly as STIX or as descriptive PDF documents, as discussed via Fig. 6. The network size increases with the increase in the numbers of hosts, links, and installed software/services installed on the hosts. Minor increases in memory reservation, inference time, and processor utilization are observed with the increase in the number of imported STIX and network instances of various sizes, ranging from 50 kB(100 hosts) to 300 kB.

In the following, we discuss the performance of the reasoning process with respect to some of the important rules in terms of time, memory consumption, and processor utilization. For all the experiments the *STIX-Analyzer* framework was deployed on an Intel (R) Core(TM) i5 machine with 2.60 GHz CPU, x-64-based processor, and 4 GB of RAM. The operating system of the machine was Microsoft Windows 8.1, 64-bit.

Table 10 – Comparative analysis of threat analytics frameworks.

	Effectiveness for threat analytics			
	Type of analytics	Inputs supported	Integrators/Support	Network analysis
STIX-Analyzer	Network relevance, impact assessment, and threat attribution.	Both structured and unstructured	STIX/TAXII, owl, swrl, SPARQL, CybOX, CVE	Yes
STIX-VIZ	Threat visualization.	Structured	STIX/TAXII	No
Soltra Edge	Threat sharing and aggregation. Creating actions.	Both structured and unstructured	STIX/TAXII, Adapters for CRITS and Snort (Soltra Solutions, 2015)	No
Spiderfoot	Penetration testing and footprinting.	None	Whois, SHODAN, RIPE, etc (SpiderFoot, 2012b).	Yes



(a) Relevance Score (S_i) and Threat Likelihood (L) (b) Quantitative (A_n) and Qualitative Asset Loss (A_l) (c) Threat Reachability (R) and Threat Actors Attribution

Fig. 9 – Relative inference time (sec) during reasoning.

5.2.1. *Time efficiency*

The efficiency of the proposed framework is measured in terms of time required by the rule engine to perform reasoning and analysis on an increasing number of ontology instances. The *Drools* rule engine (Protegeproject, 2014) is used to execute SWRL rules to perform semantic reasoning through the *Pellet* reasoner. Although numerous reasoners are available to perform inference, *Pellet* takes comparatively less time for contextual analysis, inference, and result derivation (Ulicny et al., 2014). The reasoner takes only a few seconds (measured via *Drools*) to perform inference on imported STIX and network instance dataset by executing the rules defined in *STIX-Analyzer*.

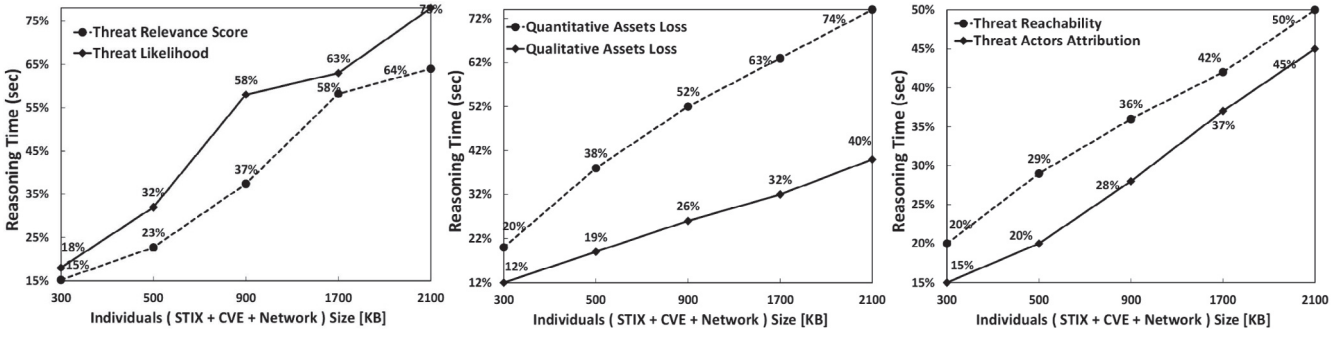
Fig. 9 shows the relative time taken by the reasoner to infer results by executing rules for RelevanceScore (S_i), Threat Likelihood (L), Quantitative Asset Loss (A_n), Qualitative Asset Loss (A_l), Threat Reachability (R), and Threat Actor Attribution on the ontology individuals (i.e., STIX, CVE, and the network). A gradual increase in time was observed for deriving results through inference while instances are being imported. Fig. 9(a) shows a relatively steeper rate of increase in the case of the inference for S_i as opposed to the inference for L. This is because the computation in the first case depends on mappings performed by nine separate sub rules (Section 3.4) which is not required in the latter case. In Fig. 9(b), it is observed that calculating *Qualitative Asset Loss* (A_l) consumes more time than calculating *Quantitative Asset Loss* (A_n). This is because all quantitative assets are calculated in A_l with respect to their assigned scale values and specified CIA preference (Section 3.4). Similarly, as shown in Fig. 9(c), the process of *Threat Actor Attribution* requires more time as compared to that of *Threat Reachability* (R) as the associated rules need to map a large number of parameters and the resulting calculation therefore is more complex than in the latter case.

5.2.2. *Processor utilization*

During the execution of inference rules, the processor is utilized by the Pellet reasoner for a short time period (mostly a few milliseconds). The reasoning process depends on the number of declared and used properties in imported instances. Fig. 10 shows the relative utilization of the processor by the *Drools* rule engine with respect to the size of declared properties. With regard to the inference time, the processor utilization for the *Threat Relevance* computation is the maximum among all defined rules.

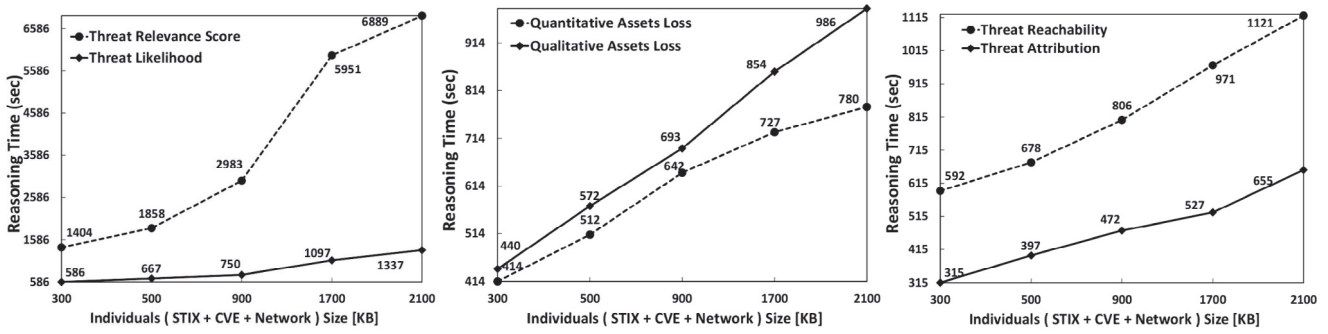
5.2.3. *Memory reservation*

All the axioms, classes, data, objects, defined and inferred properties, and instances of the ontology model consume memory. Since the proposed framework considers three different domains (STIX, network and CVEs), the resulting comprehensive ontology has a significant memory footprint. The classes, data, and object properties in the framework are static and they consume a fixed chunk of memory. While the memory size varies slightly depending on the instance size, the increase in



(a) Relevance Score (S_i) and Threat Likelihood (L) (b) Quantitative (A_n) and Qualitative Asset Loss (A_l) (c) Threat Reachability (R) and Threat Actors Attribution

Fig. 10 – Relative CPU utilization during reasoning.



(a) Relevance Score (S_i) and Threat Likelihood (L) (b) Quantitative (A_n) and Qualitative Asset loss (A_l) (c) Threat Reachability (R) and Threat Actors Attribution

Fig. 11 – Relative memory reservation ruring reasoning.

usage are mainly attributed to the number of imported instances and the inferred properties. Fig. 11 shows the relative memory consumption by *STIX-Analyzer* during the execution of the major rules by the reasoning process.

5.3. *User study*

We conducted a study to verify the effectiveness and efficiency of *STIX-Analyzer* from the user's perspective. All of the participants were IT experts familiar with the domain of cyber intelligence. The demographic summary is shown in Table 11. The participants were provided with the *STIX-Analyzer* software along with configuration details and sample threat feeds. They were asked to use the provided analytics on some sample network configurations. Afterward, the participants filled out the survey questions and provided suggestions. According to the survey results, shown in Table 12, a 100% of the survey respondents were in favor of automation for network risk assessment. In Question 2, we received feedback for improving the process of threat relevance determination with the network. The suggestions were regarding incorporating properties of network affected assets, such as usage, and state (whether encrypted, hidden, locked, and protected) for asset relevance computation. Some users proposed that the cyber-security policies of nation states can be considered, as there are higher chances of attacks being initiated from specific regions around the globe where cyber-security laws and liabilities are less stringent. Another useful suggestion had to do with using the results of the threat attribution as an input to the relevance determination. Higher risk scores may be assigned to network properties matching attacks that were seen more frequently or were easier to launch. All participants agreed that STIX-Analyzer's analytics were more accurate, easy-to-perform and efficient as compared to the manual approach. 73% of the participants received accurate and understandable results. There was no feedback on identifying competing tools in Question 6. The statistics showed that minimal effort is required for customization of reasoning logic. We plan to incorporate these suggestions in our future work.

Table 11 – Summary of demographics.

Participants detail	Count
Total users	30
Education:	
Postgraduate	30 (100%)
Expertise:	
Malware	7 (24%)
Forensics	5 (16%)
Software developer	18 (60%)
Age	25–35
Working experience	4years–8years
General computer use	9–12 hours a day
Awareness of:	
Network	30 (100%)
Threat and APTs	24 (80%)
STIX,TAXII and CyBOX	18 (60%)
Familiarity with	
Threat analytics	15 (50%)
Threat attribution	9 (30%)

6. Discussion

STIX-Analyzer has some limitations. It relies on a data-driven approach, and it is, therefore, dependent on the quality of information provided in the form of imported instances from shared threat repositories and supplied network configurations to perform threat analysis. For concrete and meaningful results, the imported instances must contain comprehensive information related to reported threats and network architectures. Incomplete information will derive partial scores and provide inaccurate impact. From our research experiments, we conclude that most of the shared STIX, and APT threat reports and network architectural data carry incomplete information. Threat analysis results will improve greatly if imported instances can be verified. Efforts by the government and industry are increasing for accepting CTI sharing not only as a standard but also as a routine or a process. Assuming that the quality of threat intelligence data will improve in the future, STIX-Analyzer's contribution of automated impact analysis of threats with respect to the network and threat profiling will grow significantly.

7. Conclusion

The proposed threat analytics framework provided a proactive mechanism for gaining insights into cyber-attacks if they

Table 12 – *STIX-Analyzer* evaluation survey.

Category	Survey questions	Results		
		Yes	Some extent	No
Effectiveness	**1.** In your view is there a current need for better automation of network risk assessment and security hardening against cyber threats?	100%	0%	0%
	2. The factors used by STIX-Analyzer for relevance identification of threats with network configurations are appropriate. If "No" what other factors would you suggest.	56%	44%	0%
	3. Profiling rules applied to multiple feeds allow for more accurate threat attribution as compared to the current manual process.	100%	0%	0%
	4. The results obtained from STIX-Analyzer are accurate and understandable.	73%	27%	0%
Efficiency	**5.** The dynamic threat analytics capability provided by STIX-Analyzer is easier to use as compared to manual analysis.	100%	0%	0%
	6. STIX-Analyzer takes comparatively less time for impact derivation and threat profiling as compared to competing tools in the market. If No, name the tool.	89%	11%	0%
	7. Modifying the reasoning logic to add additional threat analytics functionality requires minimal effort.	71%	29%	0%
	8. Automated inference as provided by STIX-Analyzer allows me to be more efficient in my job.	100%	0%	0%

were to manifest within the network by measuring their impact and associated risks. It especially took into consideration the harsh reality that the cyber threat landscape changes constantly over time, making it virtually impossible for network security administrators and incident response teams to proactively plan for the entire threat universe. The *STIX-Analyzer* framework was designed based on ontologies to capture the context of threat intelligence, network and system vulnerabilities. Appropriate rules were defined to investigate the information present in the STIX feeds, and concluded results were mapped to the organization's network architecture. To defend the network against attacks, a proactive threat detection was also devised within the proposed framework for threat actor profiling by correlating multiple STIX feeds. A detailed evaluation of the framework was conducted to analyze its effectiveness and efficiency. In the future, we plan to incorporate the valuable suggestions gathered through the user study in Section 5.3 to enhance the relevance identification mechanism. We will also design an automated mechanism to measure the validity and quality of shared threat knowledge using STIX and allow for its enrichment from various sources.

REFERENCES

Bob Gourley. Cyber threat intelligence feeds – the cyber threat. Available from: http://thecyberthreat.com/cyber-threat-intelligence-feeds/, 2014. [Accessed 16 June 2016].

Burger EW, Goodman MD, Kampanakis P, Zhu KA. Taxonomy model for cyber threat intelligence information exchange technologies. In: Proceedings of the 2014 ACM workshop on information sharing & collaborative security. ACM; 2014. pp. 51–60.

Complexible/pellet. An Open Source OWL DL reasoner. Available from: github.com/complexible/pellet. 2013. [Accessed 2 May 2015].

CO-ODE Project. OWL research at the University of Manchester. Available from: http://owl.cs.manchester.ac.uk/research/co-ode/. 2009. [Accessed 16 June 2016].

DTCC, FS-ISAC. Soltra. Available from: soltra.com/. 2015. [Accessed 24 April 2016].

Fransen F, Smulders A, Kerkdijk R. Cyber security information exchange to gain insight into the effects of cyber threats and incidents. E & I Elektrotechnik und Informationstechnik 2015;132(2):106–12.

FS-ISAC (Financial Services – Information Sharing and Analysis Center). Available from: fsisac.com/, 2015. [Accessed 24 January 2016].

FTR-Team, LUCKYCAT APT. Inside an APT Campaign with Multiple Targets in India and Japan. Available from: trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp_luckycat_redux.pdf, 2012. [Accessed 21 May 2015].

Goldstein M, Perlroth N, Sanger DE. Large, ongoing incident. FBI Investigation. Multiple banks. Available from: dealbook.nytimes.com, 2014. [Accessed 21 May 2015].

GReAT. red-october-diplomatic-cyber-attacks-investigation. Available from: securelist.com/analysis/publications/36740/red-october-diplomatic-cyber-attacks-investigation/, 2013. [Accessed 3 February 2015].

GReAT. Wild Neutron – economic espionage threat actor returns with new tricks. Available from: securelist.com/blog/research/71275/wild-neutron-economic-espionage-threat-actor-returns-with-new-tricks/, 2015. [Accessed 2 May 2015].

Hail a Taxii. Hail a TAXII. Available from: hailataxii.com/; 2015. [Accessed 24 January 2016].

Heckmann O, Piringer M, Schmitt J, Steinmetz R. On realistic network topologies for simulation. In: Proceedings of the ACM SIGCOMM workshop on models, methods and tools for reproducible network research. ACM; 2003. pp. 28–32.

Hofmann T. Probabilistic latent semantic analysis. In: Proceedings of the fifteenth conference on uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc.; 1999. pp. 289–296.

Horridge M, Drummond N, Goodwin J, Rector AL, Stevens R, Wang H. The Manchester OWL Syntax. In: OWLED, vol. 216, 2006.

InfoSec. Sony Online Entertainment. Millions of personal and credit card information and personal details were stolen. Available from: resources.infosecinstitute.com/cyber-attack-sony-pictures-much-data-breach/, 2011. [Accessed 21 May 2015].

Kaspersky-Team. The MsnMM Campaigns. Available from: securelist.com/files/2015/05/TheNaikonAPT-MsnMM1.pdf, 2015. [Accessed 21 May 2015].
Knublauch H, Fergerson RW, Noy NF, Musen MA. The protégé owl plugin: an open development environment for semantic web applications. In: The semantic web – ISWC 2004. Springer; 2004. p. 229–43.
Landauer TK, Foltz PW, Laham D. An introduction to latent semantic analysis. Discourse Process 1998;25(2–3):259–84.
Li Z, Lei J, Wang L, Li D, Ma Y. Towards identifying true threat from network security data. In: Intelligence and security informatics. Springer; 2007. p. 160–71.
libvirt. Network XML format. Available from: libvirt.org, 2015. [Accessed 21 May 2015].
McGrew D, Verma J. Making threat intelligence actionable: recommending responses with STIX. Slides of a talk presented at the RSA Conference on Analytics & Forensics, San Francisco, USA, April 23, 2015.
Medina A, Lakhina A, Matta I, Byers J. BRITE: an approach to universal topology generation. In: Proceedings of the ninth international symposium on modeling, analysis and simulation of computer and telecommunication systems. IEEE; 2001. pp. 346–353.
Mitre. STIX-Viz. Available from: github.com/STIXProject/stix-viz, 2013a. [Accessed 24 April 2016].
MITRE. STIX – Structured Threat Information Expression. Available from: stix.mitre.org/, 2013b. [Accessed 24 January 2016].
MITRE. High level use cases identified for STIX include. Available from: https://github.com/STIXProject/use-cases/wiki, 2013c. [Accessed 12 April 2016].
MITRE. Schema documentation for stix_default_vocabularies.xsd. Available from: stix.mitre.org/language/version1.0.1/xsddocs/default_vocabularies/1.0.1/stix_default_vocabularies_xsd.html, 2013d. [Accessed 12 April 2016].
MITRE. AssetTypeVocab-1.0. Available from: stixproject.github.io/data-model/1.2/stixVocabs/AssetTypeVocab-1.0/, 2013e. [Accessed 20 January 2016].
MITRE. MotivationVocab-1.1. Available from: stixproject.github.io/data-model/1.2/stixVocabs/MotivationVocab-1.1/, 2014a. [Accessed 20 January 2016].
MITRE. STIX Vocabulary-IncidentEffectVocab-1.0. Available from: stixproject.github.io/data-model/1.2/stixVocabs/IncidentEffectVocab-1.0/, 2014b. [Accessed 12 April 2016].
MITRE. MAEC – Malware Attribute Enumeration and Characterization. Available from: maec.mitre.org/, 2015a. [Accessed 2 May 2015].
MITRE. Common attack pattern enumeration and classification. Available from: capec.mitre.org, 2015b. [Accessed 2 May 2015].
MITRE. TAXII – Trusted Automated Exchange of Indicator Information. Available from: taxii.mitre.org, 2015c. [Accessed 2 May 2015].
MITRE. Cyber Observable eXpression. Available from: cybox.mitre.org/, 2015d. [Accessed 2 May 2015].
MITRE, STIX Language: Version 1.2. (Archive). Available from: stix.mitre.org/language/version1.2/, 2015e. [Accessed 2 May 2015].
NSNAM. ns-3. Available from: nsnam.org; 2011. [Accessed 2 May 2015].
NVD. CVE_FEED. Available from: nvd.nist.gov/download.cfm#CVE_FEED, 2015a. [Accessed: 2 May 2015].
NVD. National Vulnerability Database. Available from: nvd.nist.gov/download.cfm, 2015b. [Accessed 2 May 2015].
OASIS. STIX Version 1.2.1. Part 14: Vocabularies. Available from: docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part14-vocabularies/stix-v1.2.1-csprd01-part14-vocabularies.html, 2015. [Accessed 12 April 2016].
OntologWiki: Onto Clean. OntoClean. Available from: ontolog.cim3.net/cgi-bin/wiki.pl?OntoClean, 2004. [Accessed 27 January 2016].
PRISEM – Office of the Chief Information Officer. Available from: https://ocio.wa.gov/news/prisem, 2011. [Accessed 16 June 2016].
Protegeproject. protegeproject/swrlapi-drools-engine. Available from: github.com/protegeproject/swrlapi-drools-engine/wiki/SWRLDroolsTab, 2014. [Accessed 1 February 2015].
Qamar S, Anwar Z. Cyber-threat analytics. Available from: http://srg.seecs.nust.edu.pk/stix/stix-analyzer.html, 2016. [Accessed 27 January 2016].
REN-ISAC. Available from: http://www.ren-isac.net, 2008. [Accessed 16 June 2016].
Sancho D, Torre JD, Bakuei M, Villeneuve N, McArdle R. IXESHE An APT Campaign. Available from: trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp_ixeshe.pdf, 2012. [Accessed 21 May 2015].
Sans. Who's using cyberthreat intelligence and how?. Available from: https://www.sans.org/reading-room/whitepapers/analyst/cyberthreat-intelligence-how-35767, 2015. [Accessed 16 June 2016].
Sherstobitoff R, Liba I. Revealed: operation shady RAT. Available from: mcafee.com/us/resources/white-papers/wp-operation-shady-rat.pdf, 2015. [Accessed 2 May 2015].
Soltra Solutions. soltra-adapters/adapter-snort. Available from: github.com/soltra-adapters/adapter-snort, 2015. [Accessed 24 April 2016].
SpiderFoot. SpiderFoot. Available from: spiderfoot.net, 2012a. [Accessed 24 April 2016].
SpiderFoot. SpiderFoot features. Available from: spiderfoot.net/info/, 2012b. [Accessed: 24 April 2016].
Stanford. A free, open-source ontology editor and framework for building intelligent systems. Available from: protege.stanford.edu/, 2014. [Accessed 24 January 2016].
STIX. FireEye Poison Ivy Report. Available from: stixproject.github.io/examples/poison_ivy-stix-1.2.zip, 2015a. [Accessed 21 May 2015].
STIX. Givaudan's formula system. Available from: github.com/STIXProject/schemas-test/blob/master/veris/2ABBCF29-CFE3-446E-BEF9-BA3A11FB2DD8.xml, 2015b. [Accessed 2 May 2015].
STIX. Mandiant APT1 Report. Available from: stix.mitre.org/language/version1.0.1/samples/poison_ivy-stix.zip, 2015c. [Accessed 21 May 2015].
Stixproject. Identifying a threat actor profile. Available from: stixproject.github.io/documentation/idioms/identity-group/, 2015. [Accessed 2 May 2015].
STIXProject. Red October Schema. Available from: github.com/STIXProject/schemas-test/blob/a02eb29b5f655467d737f93b1bbab557083e484c/veris/4F797501-69F4-4414-BE75-B50EDCF93D6B.xml, 2015. [Accessed 2 May 2015].
STIXProject/schemas-test. STIX schemas repository. Available from: github.com/STIXProject/schemas-test/tree/master/veris, 2015. [Accessed 2 May 2015].
Stumptner M, Friedrich GE, Haselböck A. Generative constraint-based configuration of large technical systems. AI EDAM 1998;12(4):307–20.
Tantsis G, Kehagias D. Ontology evaluation – Protege Wiki. Available from: protegewiki.stanford.edu/wiki/Ontology_Evaluation, 2013. [Accessed 27 January 2016].
TAXII. taxii-discovery-service. Available from: hailataxii.com/taxii-discovery-service, 2015. [Accessed 2 May 2015].
ThreatConnect. Guide to threat intelligence platforms. Available from: go.threatconnect.com/guide-to-threat-intelligence-platform, 2012. [Accessed 2 May 2015].
ThreatQuotient. threatq. Available from: threatq.com/, 2015. [Accessed 1 May 2015].

Threatstream. Security intelligence and information sharing strategy. Available from: threatstream.com, 2014. [Accessed 24 April 2015].
TopQuadrant, Inc. TopBraid Composer-Maestro Edition (IDE). Available from: http://www.topquadrant.com/tools/IDE-topbraid-composer-maestro-edition/, 2011. [Accessed 16 June 2016].
Tsai FS, Chan KL. Detecting cyber security threats in weblogs using probabilistic models. In: Intelligence and security informatics. Springer; 2007. p. 46–57.
Ulicny BE, Moskal JJ, Kokar MM, Abe K, Smith JK. Inference and Ontologies. In: Cyber defense and situational awareness. Springer; 2014. p. 167–99.
Visitology. Intelligence Powered Defense. Available from: Visitology.com/, 2014. [Accessed 24 January 2016].
W3C. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. Available from: w3.org/Submission/SWRL/, 2014. [Accessed 24 January 2016].
Yaneza J. MalumPOS. Available from: documents.trendmicro.com/images/tex/pdf/MalumPOS%20Technical%20Brief.pdf, 2015. [Accessed 21 May 2015].
Yang D, Miao R, Wu H, Zhou Y. Product configuration knowledge modeling using ontology web language. Expert Syst Appl 2009;36(3):4399–411.
Zheng K, Li M, Jiang H. Mobile and ubiquitous systems: computing, networking, and services. In: International conference mobiquitous. Springer; 2012.

Sara Qamar completed her Masters in Science in Computer and Communications Security from the National University of Sciences and Technology, Islamabad, Pakistan, under the thesis supervision of Zahid Anwar. Sara was a research assistant in his Systems Research Group. Her research interests include Threat Intelligence and Information Sharing and the Semantic Web.

Zahid Anwar received his Ph.D. and M.S. degrees in Computer Sciences in 2008 and 2005 respectively from the *University of Illinois at Urbana-Champaign*. Zahid has worked as a software engineer and researcher at *IBM*, *Intel*, *Motorola*, *National Center for Supercomputing Applications, xFlow Research and CERN* on projects related to information security and data analytics. Zahid has academic experience working as a post-doctorate fellow at *Concordia University, Canada* and as a faculty member at the *University of North Carolina at Charlotte, USA*. At present he is an Assistant Professor at *Fontbonne University*, USA and the *National University of Sciences and Technology, Pakistan*.

Mohammad Ashiqur Rahman received his Ph.D. from the College of Computing and he Informatics, University of North Carolina at Charlotte in Spring 2015. Earlier, he received his B.S. and M.S. in computer science and engineering from Bangladesh University of Engineering and Technology in 2004 and 2007, respectively. He is currently an Assistant Professor at the Department of Computer Science, Tennessee Tech University. His primary research interest covers a wide area of computer networks and communications, especially computer and information security, risk analysis and security hardening, secure and dependable resource allocation and optimal management, and distributed and parallel computing.

Ehab Al-Shaer is a Professor and the Director of the Cyber Defense and Network Assurability (CyberDNA) Center in the School of Computing and Informatics at University of North Carolina Charlotte. His primary research areas are network security, security management, fault diagnosis, and network assurability. Prof. Al-Shaer edited/co-edited more than 10 books and book chapters, and published about 100 refereed journals and conferences papers in his area. Prof. Al-Shaer received his M.Sc. and Ph.D. in Computer Science from the Northeastern University (Boston, MA) and Old Dominion University (Norfolk, VA) in 1998 and 1994 respectively.

Bei-Tseng "Bill" Chu is a Professor of the Department of Software and Information Systems at the University of North Carolina at Charlotte. His research interests include secure software development, enterprise integration and security, and information technology education. Dr. Chu led the establishment of the information security program at UNC-Charlotte.