

Which STIX feeds are relevant to my network? : An Ontology-Driven Approach for Cyber-Threat Intelligence



By
Sara Qamar
NUST201362793MSEEC63013F

Supervisor
Dr. Zahid Anwar
Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree
of Masters in Information Security (MS IS)

In
School of Electrical Engineering and Computer Science,
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.

(April, 2016)

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

In the Name of Allāh, the Most Gracious, the Most Merciful

Approval

It is certified that the contents and form of the thesis entitled “**Which STIX feeds are relevant to my network? : An Ontology-Driven Approach for Cyber-Threat Intelligence**” submitted by **Sara Qamar** have been found satisfactory for the requirement of the degree.

Advisor: **Dr. Zahid Anwar**

Signature: _____

Date: _____

Committee Member 1: **Dr. Asad Waqar Malik**

Signature: _____

Date: _____

Committee Member 2: **Ms. Ayesha Kanwal**

Signature: _____

Date: _____

Committee Member 3: **Ms. Hirra Anwar**

Signature: _____

Date: _____

Dedication

I dedicated my research work

to

My Parents and My Sister

for their encouragement and exemplary understanding with full support to accomplish it successfully.

Certificate of Originality

I hereby declare that this submission titled **Which STIX feeds are relevant to my network? : An Ontology-Driven Approach for Cyber-Threat Intelligence** is my own work. To the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: Sara Qamar

Signature:_____

Acknowledgment

Special gratitude to Almighty Allah, Who blessed me with the vision and strength to accomplish my research.

I would like to thank my supervisor Dr.Zahid Anwar. His concern and effort provide great support and was main source of motivation for me. In his busy schedule, he was always available for consultations. He guided me during my research work and his recommendations helped me to conclude different chapters. It is because of his direction that makes my work successful. I am also very grateful to my committee members for their time and backing: Dr. Asad Waqar Malik, Ms. Ayesha Kanwal and Ms. Hirra Anwar. I am especially obliged to my Department administration to provide me such platform and for their appropriate assistance.

I would like to acknowledge my parents and siblings for their cooperation, understanding and backing they provided during the entire course of my studies.

Sara Qamar

Table of Contents

1	Introduction	2
1.1	Problem Statement	2
1.2	Motivation	3
1.3	Contribution	4
2	Literature Review	6
2.1	Background	6
2.1.1	STIX	6
2.1.2	Ontology	7
2.2	Related Work	8
3	Design and Methodology	11
3.1	STIX-Analyzer	11
3.1.1	Defining Ontologies for STIX, Network and CVEs . .	12
3.1.2	Imported Instances	19
3.1.3	Mapping Threat to Network	21
4	Analytics	23
4.1	Risk Analysis	23
4.1.1	Threat Relevance with Network (F)	25
4.1.2	Threat Likelihood (L)	31
4.1.3	Total Loss of Affected Assets (\bar{A})	33
4.1.4	Threat Reachability (R)	34
5	Case Study	36
5.0.5	Case Study: Red October STIX Impact on Network . .	36
6	Threat Profiling	41
6.1	Proactive Detection of Threats	41
6.1.1	Threat Frequency Analysis	42
6.1.2	Traffic Analysis	43

6.1.3	Threat Actors Profile	43
7	Evaluation	44
7.1	Structural Evaluation	44
7.1.1	Clarity	45
7.2	Conceptual Evaluation	46
7.2.1	Capability	46
7.2.2	Expandability, Reusability and Scalability	46
7.2.3	Performance	47
7.2.4	Reasoning Quality	50
8	Conclusion and Future Directions	55
8.1	Conclusion	55
8.2	Future Work	56
9	Appendix A : Setting up STIX-Analyzer	57
9.1	Introduction	57
9.1.1	Configuration	58
9.1.2	Installing Required Software	58
9.1.3	Installing STIX-Analyzer	59
9.1.4	Importing STIX-Analyzer Instances	61

List of Figures

3.1	STIX-Analyzer	12
3.2	Threat Analytics Ontology	14
3.3	Relations in Ontological Model	16
5.1	Network Example	37
7.1	Consistency Evaluation	46
7.2	Relative inference Time (sec) during reasoning	48
7.3	Relative CPU Utilization during reasoning	49
7.4	Relative Memory reservation during reasoning	50
7.5	Relevance Factors (F) found in STIX	52
7.6	Threat Actor's Attributes found in STIX	53
7.7	Relevance Factors (F) found in Networks	54
9.1	Protege 3.5 downloads	58
9.2	STIX-Analyzer Screen	59
9.3	Configure options	60
9.4	Select Configuration Tabs	60
9.5	OWL Classes View	61
9.6	OWL Object Properties	62
9.7	OWL Data Properties	63
9.8	Import STIX-Analyzer Instances	64
9.9	Imported Instance Attributes in Individuals Tab	64
9.10	Imported Instance Attributes in Individuals Tab	65

List of Tables

4.1	Qualifying Concepts with Notations	24
4.2	Observed Threat Relevance Elements	29
4.3	Observed Threat Relevance Elements	30
4.4	Relevance Factors (F) and associated Weights (W_i)	33
5.1	Network STIX Relevance	38
5.2	Threat Likelihood (L)	39
5.3	Quantitative Asset Cost (C_n)	39
5.4	Qualitative Asset Cost (C_l)	39
5.5	Total Loss Of Affected Assets (\bar{A})	40
7.1	Ontology Structure	45

Abstract

Automated analytics of cyber threat knowledge is crucial for network threat isolation and risk mitigation. Consequently, there has been growing interest in implementing a proactive line of defense through threats profiling. However, determining the resiliency of particular network configurations with respect to relevant threats reported in cyber threat intelligence (CTI) shared data remains a challenge, largely due to lack of semantics and contextual information present in textual representation of the threat knowledge. To overcome the limitations of existing CTI frameworks, we devise a threat analytics framework known as *STIX-Analyzer* based on Ontology Web Language (OWL) for formal specification, semantic reasoning and contextual analysis that allows the derivation of network associated threats from 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 Exposures (CVEs) for risk analysis and threat actors profiling. *STIX-Analyzer* provides an automated mechanism for realizing cyber threats targeting the network under question by classifying the threat relevance, determining threat likelihood, total loss of affected assets, threat reachability and attributing threats to their sources through formulated rules and inference. Threat attribution analyzes threat frequency, traffic and actors profile. Comprehensive structural and conceptual evaluation is performed on critical APTs/espionages from credible source on collection of arbitrary network to examine OWL clarity, consistency, capability, expandability, reusability, scalability in terms of reasoning time, memory reservation and processor utilization with the quality of reasoning achieved during threat relevance identification and threat actors attribution with the attributes present in network imported instances.

Chapter 1

Introduction

*‘Innovation Distinguishes Between A
Leader And A Follower’*

— Steve Jobs

Chapter 1 is focused on overview, motivation and background concepts used in our research related to Threat Analytics which is a rapidly evolving trend in cyber security. Key points, concepts and techniques are discussed on which rest of thesis is based on. This chapter further introduces the terminologies, keywords and paradigm used to design our proposed threat analytics framework as STIX-Analyzer. The main motivation behind the research work along with the contribution is explained in detail. The objectives, scope and theme of thesis for the development of STIX-Analyzer is summarized in this chapter. The chapter is concluded with the organization of thesis work and by emphasizing the goals of each chapter.

1.1 Problem Statement

The exponential increase in cyber-attacks with the proliferation of sophisticated hacking tactics are creating strong security concerns for the network administrators and users. As the risk impact due to cyber-attack is increasing day by day, more complex attacks on huge data with credential breaches are launched in a matter of a few hours. It is a need of the hour to automate intelligence especially for risk assessment. Traditional approaches of manually identifying, categorizing and then countering each emerging threat are not effective when dealing with a diversified and voluminous set of attack vectors in the form of APTs. Sharing of threat information between various communities via CTI frameworks has been recently gaining momentum with the

intent of creating a proactive line of defense based on knowledge of impending attacks and understanding of attacker's intentions and capabilities. STIX [1] is one such community-driven effort to develop a standardized language to define cyber threats and document their instances reported at different collaborating nodes. The information recorded using STIX is periodically shared among trusted parties using TAXII [2], which provides enhanced situational awareness regarding emerging threats with the intention that they will help in their timely and efficient neutralization. Numerous threat discovery services exist for example FS-ISAC [3] regularly maintains and distributes threat intelligence data for financial industry members around the globe. Similarly, Hail-a-Taxii [4] works in collaboration with different communities, providing CTI data as a free service with a current size of threat indicators amounting to nearly 0.3 million. Keeping in view the volume, diversity and complexity of the information reported by such services, manual threat analytics of such feeds become impractical. The main limitation of cyber threat intelligence feeds in the form of STIX is that the format is text based (XML or JSON) which is not very suited for automated analysis and context-aware reasoning. Nonetheless, such information is deemed very crucial for timely protection of critical assets through active defense strategies.

1.2 Motivation

Generally STIX feeds are generated manually, by human security analysts to share it among different communities. Limited mechanisms are available that verify or validate the usefulness of STIX information before sharing. Therefore occasionally anonymously shared CTI data might include incomplete or incorrect information. Extensive amount of STIX data is received on a daily basis and identifying its relevance and impact for a particular network is non-trivial if not impossible in many cases for a network analyst managing an enterprise. A solution is proposed that performs analytics on data obtained from existing repositories of intelligence frameworks (STIX/TAXII) to identify threat relevance on a network. The proposed methodology works by defining ontology for network, CVEs and STIX, a format suitable for automated reasoning. OWL [5] is a web ontology language based on World Wide Web Consortium (W3C) Standard. OWL is syntax independent language that can be easily interpretable by humans and machines. Knowledge and concepts represented in ontology are reusable and scalable [6]. OWL is highly expressive that allows defining concepts of wide and complex domains as compared to object oriented methods, database management systems and

generative constraint satisfaction approach [7]. OWL features allow automatic inference and semantic reasoning based on defined domain knowledge and concepts. As OWL is recommended by W3C, it can be incorporated and assembled with other CTI frameworks and tools. We populate our ontology model both with elements extracted from descriptions of emerging threats such as Tactics Techniques and Procedures (TTPs), indicators, observables and exploit targets as well as elements in the network. Network ontology is designed to analyze threat data on computer networks and real 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) [8] that operate on our ontology model and perform mapping of the threat, vulnerabilities and network elements.

1.3 Contribution

The *STIX-Analyzer* improves the capability of timely threat and risk identification on network, aims 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 to relate the shared threats knowledge with network architectural knowledge and analyzes the threat relevance with network. Our proposed framework is based on ontology that has three major domains; STIX, Network and CVEs. The devised *STIX-Analyzer* ontology has provision for run time data import, supports multiple form of data stores including csv, excel, xml and Jena. The reasoning process is independent to the underlying procedures used to maintain repositories for threats, network and vulnerabilities. The designed ontology model and rules not only identifies the critical threats and vulnerabilities of network but also act proactively against attack threats, provide essence of threats on network before actual attack triggers and update the firewall policy for targeted host on network. Our ontology determines the victim's network targeted by STIX feeds, computes its likelihood, its reachability to network host and identifies the associated risk impact on network through inference performed by defined rules. To automatically identify the likelihood of a network breach and its resulting impact if the threat or its variation were to manifest in our network. The defined rules and network information provide support for identifying the exploitable path reachable from threats to hosts. Similarly, *STIX-Analyzer* helped us in identifying the threat source by analyzing the patterns of attack campaigns. Rules are formulated to design a proactive strategy for attack

identification before its occurrence on network. A comprehensive evaluation on proposed *STIX-Analyzer* is performed in two ways; structural evaluation and conceptual evaluation. Structural evaluation is executed to check threat domain coverage comprises of ontology structure, clarity and consistency among various attributes and relations. Ontology conceptual evaluation is achieved by measuring ontology capability, reusability, scalability and expandability, and performance in terms of efficiency, processor utilization and memory reservation. Quality of reasoning is evaluated for threat relevance, actors attribution and network architecture elements.

The remaining thesis is organized as follows. In section 2.1, background regarding proposed ontology and tools used to perform reasoning are discussed. In section 2.2, related research and few existing related platforms are briefly described. In section 3.1, proposed methodology is explained for the network, STIX, CVE with the corresponding description of analyzing threats on the network for performing attack relevance and risk assessment. Section 6.1 provides proactive threat detection and threat actor's attribution. Complete evaluation and of *STIX-Analyzer* is performed in section 7. In the proceeding sections, we conclude this thesis with *future work*.

Chapter 2

Literature Review

‘We all need people who will give us feedback. That’s how we improve.’

— Bill Gates

Chapter 2 explains the existing CTI trends and framework for analyzing threats on network. Major contributors in the domain of threat analytics, commercial endeavor and literature survey of existing tools and techniques to identify risk impact of threats on network are discussed in this section. Limitations of existing mechanism with the importance of proposed Threat analytics framework that aims for automated, actionable analytics, contrary to the traditional manual analysis of threats are elaborated.

2.1 Background

Here in this section, we briefly discuss background concepts related to our proposed threat analytics ontological solution.

2.1.1 STIX

Different communities are collaborating to cater the increased number of cyber threats and their corresponding attacks. STIX [1] framework is a noteworthy and novel effort in this regard and plays its role by defining a high level schema document to map different attack patterns and related threats. It works in collaboration with CybOX, Common Attack Pattern Enumeration and Classification (CAPEC) [9] and Malware Attribute Enumeration

and Characterization (MAEC) [10] for providing attack and malware details and TAXII for exchange of cyber information. STIX serves as all in one package for dealing with cyber threat information. 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, where the incident happened and associated campaign and finally what should be the course of actions. STIX is shared via TAXII that defines a set of services, message types and message exchanges. The messages are represented in a particular format e.g. XML but is not limited to any specific language binding. Further the messages are carried over the network by pre-existing network protocols such as HTTP/ HTTPS. Multiple network protocols can be used depending on the requirement.

2.1.2 Ontology

The term ‘Ontology’ stands for a mathematical, logical, formal, machine readable model with semantic meaning. The ontology constructs are classes, sub-classes, properties mapped through relations, restriction, constraints and instances of classes. OWL holds two type of properties, (i) object and (ii) data properties. Object properties relate one member with others and data properties relates 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 class. Restrictions such as ‘equals’, ‘some’ (some values from range), ‘has’ (at-least one value) and ‘at most’ are used to hold relationships between the members of a class. Properties are explained by specifying domain and range characteristics. Different OWL editors are available, we have chosen Protege 3.5 [11], because it is an open source with community support and extensive features. Protege offers extensive support for various plugins for visualization, OWL conversion, database storage and reasoning. Different reasoners are also available. We preferred to use Pellet [12] because of its provision in SWRL and SWRL Built-In rule development and execution. SWRL[8] is used to describe rules and logic as it also has a human and machine interpretable simple syntax. SWRL built-in rules provide built in functions for strings, URIs, mathematical formulas, e.g., ‘add’, ‘equal’, ‘substring’, and many more. Protege provides *SWRLDroolsTab* [13] as a *Drools rule engine* [13] that executes the SWRL rules in Protege and provides updates for inferred values. Pellet is an open source, java based description logic reasoner that helps in measuring the consistency of ontology classes and instances hierarchy, performs reasoning and computes results. Pellet also support SPARQL

[14], an RDF query language that is used in rules to derive results.

2.2 Related Work

Our work benefits from existing works on formal semantic models and contextual reasoning describing structured and unstructured threat information for network vulnerabilities analysis. We were able to find both fundamental research relevant to our work and some recently launched commercial CTI endeavors.

Tsai et al.[15] have analyzed cyber threat intelligence when the standards for representing threat information were not there. Before the emergence of CTI standards, security experts and researchers posted their cyber security threat findings and observations on web blogs. This work analyzes weblog posts for various categories of cyber security threats related to the detection of cyber-attacks, crime and terrorism. Latent Semantic analysis (LSA) is used to find semantically related topics in web blog corpus. Further important keywords of each topic are assigned quantitative measure through Probabilistic LSA (PLSA). The results proof the approach to be for broadly searching security related news in massive web blogs. The major limitation of this approach is web blogs can't model real time threat scenario as observed and experienced by an organization in the form of security data collection from security devices and tools. Also web blogs usually do not cover fine grained details regarding a threat and its mitigation that result in complete technical understanding of security buzz term/topic by the administrator himself.

Lei et al. [16] focuses on the problem of true threat identification in a distributed environment where network security data is managed at distributed locations. The system works by calculating threat score based on alert correlation. Further incidents are ranked according to threat scores. This technique is called *Alert Rank*. With an alert input, the output is calculated as four major attributes: 1) Priority Score Specification 2) Reliability Formulation 3) Asset Specification 4) Alert Threat Formulation. The proposed approach provides means of finding correlation between alerts arriving from distributed components. It can model real time threat scenarios as observed and experienced by an organization. The major limitation of this work is the lack of standardization as alert data needs to be brought in a uniform representation as devices and tools are from heterogeneous vendors.

Lack of standardization causes delay.

ThreatConnect[17] offers a Threat Intelligence Platform (TIP). It collects data from multiple sources to perform analysis and looks for indicators and their associations with other entities such as adversaries, signatures and incidents. ThreatConnect provides support for existing tools like Whois, Snort, Nesus and has also partnered with Maltego [18] [19]. A complicated process is involved that requires strong user involvement for tool selection and threat investigation based on threat reports. It categorizes threats, e.g., DDOS and web application attacks but no categorization is performed on the basis of the network specification.

ThreatStream OPTIC is a cyber-threat intelligence platform, which analyzes threats from different sources, ranks each indicator and defines relationships with known threats [20]. While the internals of the framework are not disclosed the primary focus is on using big data indexing techniques to provide fast search of large security reports. ThreatStream integrates other tools like STIX/TAXII, SPLUNK, Whois, Hash search, etc. to identify and analyze threat behavior and act accordingly. ThreatStream requires developers or skilled professionals to analyze and prioritize threats, the prioritization is measured between threats on the basis of indicators. It does not prioritize threats on the network entities or by identifying the network vulnerabilities.

ThreatQ is a threat intelligence platform (TIP) [21] that manages intelligence in a central repository, prioritizes risky threats. Its objective is to provide users a central threat repository. The role of the user is to perform analytics on derived information manually or by using some tools. It does not support any reasoning or automated analysis on centralized threats.

CISCO is developing techniques [22] for recommending actions in STIX which can be reviewed and recommended by human operators. These techniques rely on a threat intelligence aggregator and 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. Normalization and threat enrichment is required with input parameters to perform analysis. CISCO features are more focused towards threat remediation instead of threat relevance identification and threat attribution.

STIX is currently being supported and researched by many communities.

Fransen et al. [23] in their paper analyzed the timely gain of information regarding incidents from cyber security information sharing, proposing STIX for cyber situational awareness and use STIX vocabulary to enumerate impact regarding data. A detailed layered taxonomy model is presented by Burger et al. [24] to analyze CTI exchange and classifying cyber security terms. Chapter on ‘Inference and Ontologies’ [25] provides motivation for the conversion of existing cyber threat knowledge and standards into ontology and proposed ontology for STIX [26]. Identifying needs for STIX automation, as STIX can’t be loaded directly into any system without resolving interoperability issue and it requires a sophisticated parser to fetch and use its information. This chapter identifies the possible problems of STIX schema document into ontology conversion, STIX xml structure is quite complex and its constraints and restrictions cannot be converted into owl using available tools, and that XML does not support inference. It’s highly required to convert the STIX into ontology and design rules to perform inference. Proposed Visitology ontology has reasoning limitations, as no methodology is explained to perform inference on designed ontology for network mapping and risk assessment. No rules and reasoning strategy is discussed for threat attribution.

Chapter 3

Design and Methodology

‘Design is not just what it looks like and feels like. Design is how it works.’

— Steve Jobs

Chapter 3 presents the architectural overview of threat analytics framework that we design to analyze rapidly growing threats and attack vectors on network. This chapter introduces framework ontology models proposed for STIX/threat data, network and CVE. Ontology attributes, properties, relations, restrictions and instances are used to elaborate threat mapping for various network architectures with number of exploits and vulnerabilities present on network nodes.

3.1 STIX-Analyzer

In this section, we cover the details of our proposed *STIX-Analyzer* as shown in Figure 3.1. We divided our framework into six sections according to the nature of tasks performed by each section. These tasks are: defining ontologies (3.1.1) for *STIX*, *network* and *CVEs*, importing ontology instances (3.1.2), mapping threat to network (3.1.3), performing risk analysis (4.1), deriving risk impact via a case study (5.0.5), proactive threat detection via threat attribution (6.1). A complete working of these tasks is discussed in the subsequent sections.

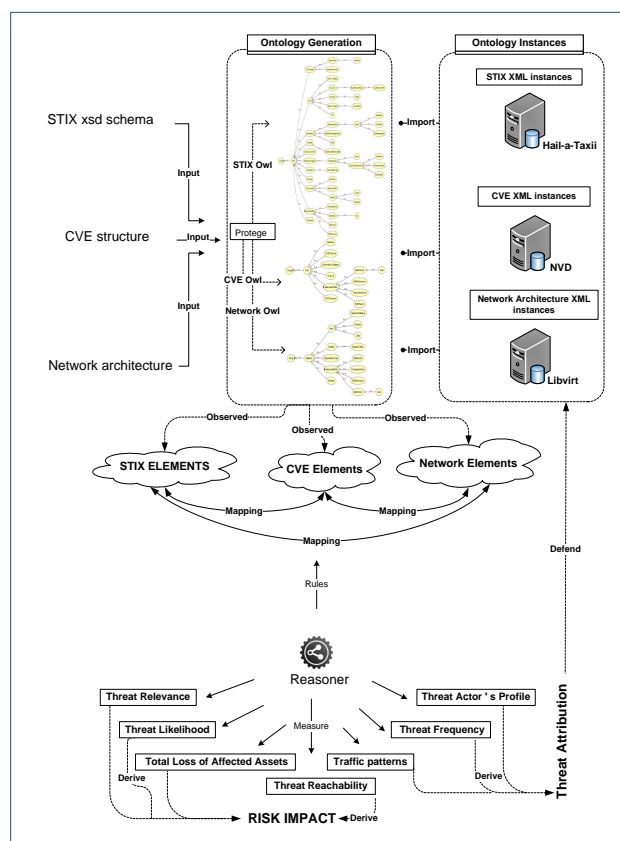


Figure 3.1: STIX-Analyzer

3.1.1 Defining Ontologies for STIX, Network and CVEs

Ontology for *STIX*, *network* and *CVEs* is defined by assembling information from three different sources i.e. *STIX xsd schema document* [27], *CVE* [28] description along with identifiers and real world network architecture model. The STIX schema document contains xml elements and attributes that represents threat knowledge. Similarly, the NVD structure for maintaining CVEs and CVSS has attributes related to CVE description and identifiers. Our real word network architecture is made up of many entities and elements such as subnets, firewall, hosts, network identity and links between connected hosts. The raw assembled information was first analyzed then we performed a few data cleansing steps necessary before ontology creation. Ontology is designed by creating classes, objects, data properties and relations corresponding to the gathered information. We have followed OWL-Manchester syntax to de-

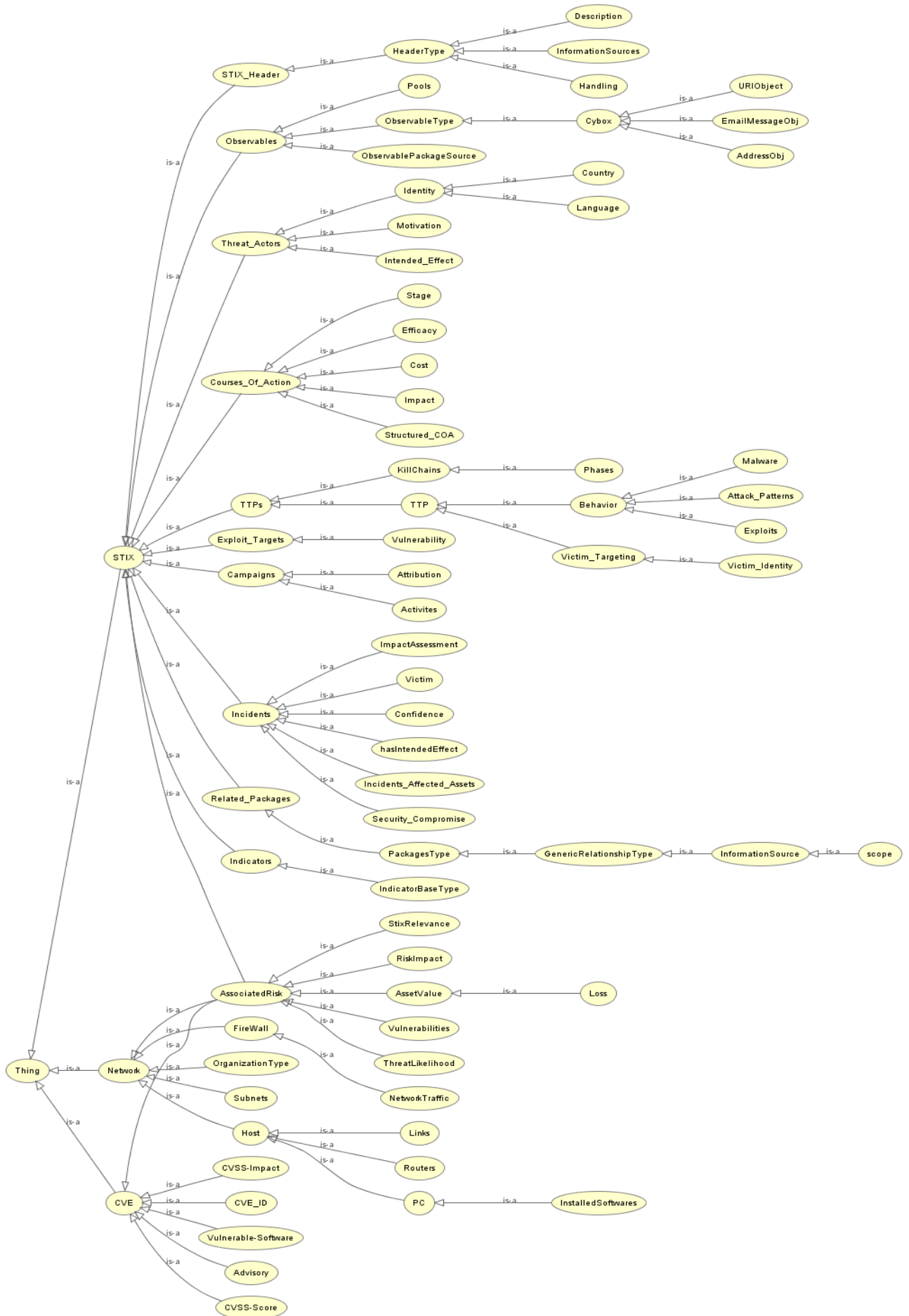
velop our ontology; that offers readable, clear, and self-explanatory terms for class hierarchy, list of data type object properties, relations (is-a, has-a) with restrictions and rules. The ontology classes are further divided into sub-classes having properly defined domains and ranges for each property. After the design step, instances are imported in the developed ontology model.

The proposed threat analytics ontology acts as a working model for STIX framework, the targeted network, and CVE score dictionary. Every ontology has root class *Thing* by default and all classes are derived from *Thing*. Proposed threat analytics ontology model *STIX*, *network* and *CVE* are inherited from *Thing* concept. High level ontology class view is generated using Protege **OWLViz** [29] tab, shown in Figure 3.2 and provided relations are shown in Figure 3.3 using Protege **Jambalaya** [30] plugin. A detailed discussion of the model is provided further below.

3.1.1.1 STIX Ontology

The complete ontology for STIX is built as proposed in its schema documents [31]. The proposed concepts of STIX ontology comprises of *Observables*, *Indicators*, *Incidents*, *TTPs*, *Exploit_Targets*, *Campaigns*, *Threat_Actors* and *Course_of_Actions*. These concepts are declared as separate classes in STIX ontology and each class is further divided into sub-classes, data types and object properties. In our proposed ontology the defined data and object properties of STIX exceeds 800. Therefore here in this thesis we have shown a few important concepts and properties of STIX model in Figure 3.2. Each one of them is briefly described below.

- **STIX_Header** contains an important data property. *STIX id*, which is unique for shared threats. The header also contains a time stamp as a reference of specific threat description.
- **Observables** describes STIX observables with the help of a sub class *ObservableType* which contains *Cybox*. *Cybox* defines observables such as *AddressObj*, *EmailMessageObj*, *URIObject* using *hasProperties* object relation.
- **Threat_Actors** reveals information about threat source's identity, motivation and its intended effect. Identity is further described via threat actor's country and language. The subclasses are *Identity*, *Intended_Effect* and *Motivation*. Identity has a relation of *hasIdentity* mapped to data



properties for *hasParty_OrganizationName*, *hasElectronicAddressIdentifier* and *hasAddress_CountryName* and has various important concepts like *hasThreat_Actor_Sophistication*, *hasConfidence*.

- **Course_of_Actions** describes the current status or level of threat with the help of *Stage*, *Efficacy*, *Cost*, *Impact* and *Structured_COA* subclasses.
- **TTPs** describes the tactics, techniques and procedures used to launch attack such as the phase of kill chain intrusion model, malware behavior and the victim. The subclass of *KillChains* is connected via *hasTTPsType* object property. Further *TTP* comprises of *Exploit_Targets*, *Victim_Targeting* and *Behavior* and has object properties like TTP has relation of *has_RelatedTTPs* and *KillChains* has property of *hasPhases* and many more.
- **Exploit_Targets** represents vulnerability as a subclass containing *CVE_id* data property using *hasCVEs* relation.
- **Campaigns** attributes threat to a particular actor. The ontology describes *Attribution* and *Activites* properties with the relation of *hasAttributed_Threat_Actor* and is mapped to *hasRelated_Threat_Actor_Type*.
- **Incidents** has many important object relations used for threat relevance (S_i) identification like *hasIdentityType* connected with the victim identity data value and many other relations are defined like *hasAffectedAssetsType*, *hasImpactAssessmentType*, *hasStatementType* (for *IntendedEffect*), *hasControlledVocabularyStringType* (for *Security_Compromise*) and *hasConfidence*.
- **Related_Packages** contains information about related STIX packages.
- **Indicators** describes pattern for observed attack using CybOX, has data properties for *Hashes* and *Signatures*.

STIX Ontology Relations and Restrictions The relations between different concepts and with itself are mapped on the ontology by identifying its domain, range and restrictions. Restrictions for *Indicator* are presented by specifying domains and ranges as shown in Listing 3.1. The domain of an indicator is *hasObservable* and its range is *Observable*. Two properties of restriction used here are: *has* (\in) and *some* (\exists). An *Indicator* must own


```

Indicator  $\equiv$  STIX  $\cup$ 
     $\in$  hasObservable has Observables  $\cup$ 
     $\exists$  hasIndicators some Indicators  $\cup$ 
     $\exists$  hasRelatedTTP some TTP

```

Listing 3.2: STIX Indicator Restrictions as DL Rule

3.1.1.2 Network Ontology

The ontology modeling for the network includes *Hosts*, *Firewall*, *Subnets*, *OrganizationType*. *Hosts* class comprises of *PC*, *routers*, *InstalledSoftware* and *Links*. Some relations between network entities (e.g., Link between PC and Path) are defined like *hasSource*, *hasRouter*, *hasPath*, *hasReachableNode* and few data properties are derived through reasoning such as *hasVulnerableSoftware*, *hasTargetedAssets*, *hasTargetedOrgRelevance* etc. Different instances of the network are modeled with complete specifications like versions of installed software, operating system installed, configured hardware and environments. This modeling is used to demonstrate and analyze the behavior of threats and exploits on various network architectures.

Network Ontology Relations and Restrictions Network ontology *Host* is a subclass of *Network* and has one or more *Firewall* configured shown by *some* (\exists) restriction property. *Host* must be connected to at least one *Host or Router*, containing a *Name*, *IP*, *SubnetID* using *min* (\geq) restrictions. Some restrictions and relations are inferred from the pre-defined rules; e.g. a host knows only about its own *InstalledSoftware*. Further rules identify the vulnerable software from the installed ones, using *some* property as shown in Listing 3.3. Some restrictions of firewall are given in Listing 3.4. *Firewall* is a subclass of *Network* and generates alerts for *Threat_Actors* mapped to ontology using *hasAlert* property with *some* restrictions because alerts are generated for one or more *Threat_Actors*. Firewall *hasAccess* to *Network-Traffic* and take at least one *Action* i.e. allow or deny. Some restrictions are used to identify the reachability of threats to network hosts by analyzing the *hasHostSrc*, *hasHostDest*, *hasNextHop* and access is allowed through firewall. Restrictions force the firewall to be configured for at-least one source, destination and next hop defined using *min* (\geq) restrictions.

```

Host  $\equiv$  Network  $\cup$ 
   $\exists$  hasFirewall some Firewall  $\cup$ 
   $\geq$  hasConnectedHost some Host  $\cup$ 
   $\geq$  hasRouter min 1  $\cup$ 
   $\geq$  hasHostName min 1  $\cup$ 
   $\geq$  hasHostIP min 1  $\cup$ 
   $\exists$  hasVulnerableSoftware some
      InstalledSoftware

```

Listing 3.3: Network Host Restrictions as DL Rule

```

Firewall  $\equiv$  Network  $\cup$ 
   $\exists$  hasAlerts some Threat_Actors  $\cup$ 
   $\exists$  hasAccess some NetworkTraffic  $\cup$ 
   $\geq$  hasAction min 1  $\cup$ 
   $\geq$  hasHostSrc min 1  $\cup$ 
   $\geq$  hasHostDest min 1  $\cup$ 
   $\geq$  hasNextHop min 1

```

Listing 3.4: Network Firewall Restrictions as DL Rule

3.1.1.3 CVE Ontology

The design of CVE ontology model is based on NVD scoring system for known vulnerabilities [32]. CVE class has *CVE_ID*, *Vulnerable-Software*, *CVSS-Score*, *Advisory*, and *Impact* sub classes.

CVE Ontology Relations and Restrictions Few restrictions are mapped to CVE class as shown in Listing 3.5. CVE class is derived from root *owl Thing* class. A single instance of CVE at most contains one CVE id, CVSS score and impact shown using *max* \leq restriction for *hasCVE_ID*, *hasCVSS_Score*, *hasImpact* properties. CVE has corresponding *hasVulnerableSoftware*, *hasAdvisory* property with at least one restriction shown using *min* \geq restriction.

```

CVE  $\equiv$  owl:Thing  $\cup$ 
   $\leq$  hasCVE_ID max 1  $\cup$ 
   $\leq$  hasCVSS_baseScore max 1  $\cup$ 
   $\geq$  hasVulnerableSoftware min 1  $\cup$ 
   $\geq$  hasAdvisory min 1

```

Listing 3.5: CVE Restrictions as DL Rule

3.1.1.4 Associated Risk

AssociatedRisk class is derived from *STIX*, *Network* and *CVE* that carries the derived result of reasoning performed for *Impact* and *RelevanceScore* computation rules discussed in subsequent sections. *AssociatedRisk* class has *RiskImpact*, *StixRelevance*, *ThreatLikelihood*, *Vulnerabilities* and *Asset-Values* sub classes. Some properties are used to compute STIX feed relevance with network. These properties are derived by executing *vulnerability identification*, *software relevance*, *attacker's motivation*, *victim's business type*, *targeted location*, *affected assets*, *CIA relevance*, *incident security relevance*, *hasVictimLanguageRelevance* rules. These rules are stored in *AssociatedRisk* class. A high level view of the mapped relations that are used to compute rules and derive results is provided in Figure 3.3. As a reference, we have highlighted some relations and objects used in Listing 4.1 to compute *STIX* relevance with network on the basis of installed vulnerable software in Figure 3.3. *STIX Exploit-Targets* contains vulnerabilities or CVE.IDs shown at the top of figure, where *STIX Exploit-Targets* is linked to *Vulnerability* using *hasVulnerability* relation. *Vulnerability* is connected to *CVE.ID* linked with *VulnerableSoftware-list* (list of vulnerable software products) using *hasVulnerableSoftware-list* property. Network *Hosts* are connected using *hasConnected* relation and the configured software on *Hosts* is depicted using *hasInstalledSoftware*, the *Hosts installedSoftware* are compared with *CVE VulnerableSoftware* list to identify vulnerable software of network.

3.1.2 Imported Instances

The designed ontology maps structured threat information on the network by populating different instances of network and STIX along with the vulnerability scores obtained from CVEs. The instances are used to perform risk analysis on real and dynamic network environment. STIX feeds are available at STIX repositories [33, 34, 32]. These feeds are used to enrich and populate *STIX-Analyzer* by importing its instances in to the STIX ontology model. Similarly, network architecture knowledge in XML format is used as Network ontology instances. We imported real network topologies as instances (generated from BRITE Topology Generator [35] and some from libvirt network [36]) into our ontology model in order to analyze threats. The complete repository of CVEs and CVSS score is available [28] on NVD website, as vulnerability feeds. It is open source in XML format. These repositories are imported into ontology with complete vulnerability information as instances of CVE class to identify network related vulnerability. To import the in-

stances, STIX feeds are enriched and minor cleaning is performed on CVEs and network XML document. As a reference few instance elements are given in Listing 3.6, 3.7 and 3.8.

To identify the vulnerabilities that are present in a network, we have imported CVE repositories into our ontology as a CVE model. We have added the vulnerabilities as CVE instances. These instances include the records of CVE impacts, CVE scores, etc. that are used in identifying threat probability.

STIX *Exploit_Targets* is shown in Listing 3.6, where *Exploit_Targets* element identifies various *Vulnerabilities* with corresponding *CVE-ID*.

```

<stix:Exploit_Targets>
  <Vulnerability>
    <CVE_ID> CVE-2009-3129 </CVE_ID>
  </Vulnerability>
  <Vulnerability>
    <CVE_ID> CVE-2010-3333 </CVE_ID>
  </Vulnerability>
  <Vulnerability>
    <CVE_ID> CVE-2012-0158 </CVE_ID>
  </Vulnerability>
  <Vulnerability>
    <CVE_ID> CVE-2011-3544 </CVE_ID>
  </Vulnerability>
</stix:Exploit_Targets>

```

Listing 3.6: STIX Exploit_Targets as XML Instance

Some elements of network imported instance are shown in Listing 3.7 containing host name and installed software. CVE instance as an example is

```

<Network>
  <hasHost_name> Host-a </hasHost_name>
  <hasinstalled-software>
    Microsoft Office 2003 SP3
  </hasinstalled-software>
</Network>

```

Listing 3.7: Network Host as XML Instance

shown in Listing 3.8, where *cve-id* is CVE-2010-3333, *cvss-score* of vulnerability is 10 and *vulnerable-software-list* includes Microsoft Office 2003 SP3, Microsoft Office 2007 SP2, Microsoft Office 2004 SP3 and Microsoft Office 2010.

```

<CVE>
  <cve-id> CVE-2010-3333 </cve-id>
  <cvss-score> 10 </cvss-score>
  <vulnerable-software-list>
    <product>
      Microsoft Office 2003 SP3
    </product>
    <product>
      Microsoft Office 2007 SP2
    </product>
    <product>
      Microsoft Office 2004 SP3
    </product>
    <product>
      Microsoft Office 2010
    </product>
  </vulnerable-software-list>
</CVE>

```

Listing 3.8: CVE as XML Instance

3.1.3 Mapping Threat to Network

STIX ontology is very complex as it comprises of more than eight hundred data types, object properties and relations. It provides extensive threat knowledge but identifying those elements of STIX that are comparable to network architecture is a challenging task. Similarly network architectural knowledge is very vast, comprises of thousands of network entities and identifying its relevance with threat/attacks knowledge is difficult. Nine elements *F* or attributes are identified in STIX and Network that are comparable and helpful in identifying *Impact*. STIX has *Exploit_Targets Vulnerability*, comparable with *CVE* and helped in identifying the vulnerable and exploitable *InstalledSoftware* in network. STIX attacker *Motivation* is mapped to network intent, Similarly the STIX *Targeted_Organization_Type* is matched with Network *Organization_Type*. STIX *Victim_Location* or *Victim_Address* or *Targeted_Country* is comparable to network *Organization_Address* or location using geo location. If the *Affected_Assets* or targeted assets are same as network *Configured_Assets* then the network is vulnerable to that threat. Some threats target the users belongs to specific culture or by identifying their mode of speech, STIX element of *Targeted_Language* is mapped to network configured software language. STIX *CIA_Affected_Property* and network *CIA_Preference* is related to identified relevant threats. *Incident_Security_Compromise* element of STIX helps in identifying the critical and completed threats. The associated rules are defined in section (4.1.1).

Chapter 4

Analytics

‘The goal is to turn data into information, and information into insight.’

— Carly Fiorina

Chapter 4 discusses the designed rules to automate relevance identification of dynamic threats on network with the derivation of risk impact and proactive threat actors attribution. Nine relevance factors are identified to relate the knowledge of STIX/APTs with network. After relevance computation, the devised rules are executed to identify threat likelihood, assets loss and threat reachability which computes the impact of threat on network.

4.1 Risk Analysis

Complete *threat analytics* is performed by *STIX-Analyzer* through defined rules. Notations are assigned to few major concepts used in defined equations and rules, given in the Table 4.1 with their definitions.

We performed risk analysis to derive *risk impact* denoted as ***I*** with the help of four parameters that we termed as four Ts of threat analytics. These four Ts are: (i) *Threat Relevance* (ii) *Threat Likelihood* (iii) *Total Loss of Affected Assets* and (iv) *Threat Reachability*. We denote these four Ts as '***F***', '***L***', '***A***' and '***R***' respectively. We have defined rules for each one of them. Threat relevance depends on a number of factors discussed later. We weigh each factor with a relevance score denoted as S_i . Threat likelihood depends on relevance scores. It is measured using the identified relevance factors as discussed in section (4.1.1). These factors influence the likelihood of threat occurrence. For instance, if the chance of fulfilling the motivation

Table 4.1: Qualifying Concepts with Notations

Notations	Definition
F	<i>Relevance Factors</i> are the major identified characteristics, used to relate threat with the network. Nine F_i are identified based on the existing knowledge of STIX, CVE and Network.
E	Each F_i has <i>Set of Sub Attributes</i> or <i>Set of Sub Elements</i> that specifies its characteristics. E_s Symbolizes E present in received STIX, E_n for received Network and the \bar{E} for all available E .
S	Each F_i has a <i>Relevance Score</i> , computed on E . S_i is derived by comparing the similarities between threats target and 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	<i>Threat Likelihood</i> provides the score for the possibility of occurrences of certain nature of threats/STIX on Network. Computed on S_i and W_i .
C	Network comprises of quantitative (servers, computers, mobile etc.) and qualitative (confidentiality, integrity and availability) assets with their associated <i>Cost</i> . C_n denotes the <i>Affected Quantitative Assets Cost</i> on Network, targeted by STIX and \bar{C}_n is for <i>Total Quantitative Assets Cost</i> available on Network. The C_l represents the <i>Affected Qualitative Assets Cost</i> on Network, damaged by threat/STIX and \bar{C}_l is for <i>Total Qualitative Assets Cost</i> available on Network.
\bar{A}	<i>Total Loss of Affected Assets</i> is a collective score that measures the maximum loss of network assets, targeted by STIX. \bar{A} is derived from <i>Quantitative Assets Loss</i> (A_n) and <i>Qualitative Assets loss</i> (A_l). A_n score is measured by dividing C_n with \bar{C}_n and A_l derivation is based on C_l & \bar{C}_l .
V	<i>Scale Value</i> assigns a criticality scale[0-100] level/value for affected qualitative assets in terms for <i>Confidentiality</i> , <i>Integrity</i> and <i>Availability</i> w.r.t Network. The A_l computation is based on assigned V .
R	<i>Threat Reachability</i> measures the accessibility of threats and attacks infiltration to network vulnerable hosts which are directly and indirectly connected through internet.
I	<i>Risk Impact</i> identifies the risky threats/STIX which has high impact on network, derived from L , \bar{A} and R .

behind launching attacks on a particular network is high, or if the network lies in a targeted country and the nature of organization matches with the type of attackers' victim, the likelihood increases. We have calculated loss both quantitatively and qualitatively denoted as A_n and A_l . Reachability are measure of the number of hosts victimized by the threat or exploitation process in the network. We have defined comprehensive rules to automate the whole risk analysis process based on the formula as shown in equation 4.1. Each term in equation 4.1 is discussed in subsequent sections along with associated rules.

$$\mathbf{I} = \mathbf{L} \times \bar{\mathbf{A}} \times \mathbf{R} \quad (4.1)$$

4.1.1 Threat Relevance with Network (F)

STIX's attributes assist in relating threats with network model under consideration. Multiple relevance factors are identified that help in determining relevance of a particular threat to a network. These factors include *vulnerable software, attacker's motivation, location, targeted language, business type, affected assets and affected CIA property and incident severity*. We have explained all these factors with reference to the defined rules of our *STIX-Analyzer's* ontology. Here in this thesis to minimize complexity, we have mentioned only a few significant *mathitSWRL* rules. We have designed equations for rules to represent the concept of reasoning performed on STIX feeds to analyze threats.

Relevance with Software Vulnerability (*hasCVE_Relevance*) Vulnerabilities are exploited to launch attacks and to compromise networks, For reference *CVEs* used in famous *Red October, LUCKYCAT, WildNeutron and Naikon* are listed in Table:4.3. In Listing 4.1, relevance is identified on the basis of vulnerabilities detected from the list of configured services and applications running on the network hosts. The rule extracts the vulnerabilities or CVE ids from the *Exploit_Target* element of STIX framework and those vulnerabilities are then mapped with the CVE ontology to identify the associated software vulnerability from the list. The CVEs of identified vulnerable software are then compared with the software installed on network hosts to detect its presence. If the identified STIX vulnerabilities exist on the network, the respective relevance score S_i i.e. *hasCVE_Relevance* for single CVE element will be set to true.

Relevance with Attackers' Motivation (*hasMotivationRelevance*) Relevance with attacker's motivation is determined by analyzing attacker's intention and motivation. Mostly, the motivations behind major cyber attacks

```

STIX (?X) ∧
  hasExploit_Targets (?X, ?e) ∧
  hasVulnerability (?e, ?v) ∧
  has_CVE-ID (?v, ?c) ∧
CVE (?C) ∧
  has_CVE-ID (?C, ?cve) ∧
  swrlb:containsIgnoreCase (?c, ?cve) ∧
  hasVulnerableSoftwareList (?C, ?s) ∧
Network (?N) ∧
  hasHost (?N, ?h) ∧
  hasHostInstalledSoftware (?h, ?y) ∧
  swrlb:containsIgnoreCase (?y, ?s) ∧
  hasAssociatedRisk (?X, ?r) ∧
  hasSTIXRelevance (?r, ?rlv)
→ hasVulnerableHost (?N, ?h) ∧
  hasHostVulnerableSoftware (?h, ?s) ∧
  hasCVE_Relevance (?rlv, 1)

```

Listing 4.1: Vulnerability Relevance as SWRL Rule

are financial gains, economic data-breaches and publicity. The relevance can be determined by comparing organizations' high objectives like financial gain, economic benefit, public accessibility and big data storage with attacker's motivation as seen in Table:4.3 where the motivation of *Red October* campaign is *Espionage*, *LUCKYCAT* campaign is *political*, *WildNeutron* APT is *Ego* and *Economic*, and the motivation of *Naikon* APT is *Ideological*. By identifying the type of the network design, we can map the attacker's motivation on the network, as financial organization or banks involve payment, withdrawal, and other money transaction services can easily find relevant threats or can be targeted by STIX where threat actor's motivation is related to *finance* or any other *economic activity*. Suppose the client network is offering online entertainment services which consume high traffic and require more upload/download storage space. It can be victimized by such a threat in which the attacker's motivation is to gain *publicity*. Attackers with such motives hacks crafted, famous, or publicly accessible websites to launch their propaganda through defacement attacks.

Relevance with Business Type (*hasOrganizationRelevance*) Network architecture varies from domain to domain. The network architecture for hotel is different from home, while the library or laboratory network design is different from banks. Relevance is also derived on the basis of targeted victim's industry type information present in TTP element of STIX. Few elements of affected industry type and organization by APTs and campaigns are shown in Table4.3 includes health sector, banking sector, military, embassies

and law firms. The targeted industry type is compared with *Organization-Type* element of network to compute relevance. Similarly, the *Incident* element of STIX also includes *Threat_Actor's* and *Victim's* organization name and administrative area that can help compute relevance.

Relevance with Target's Location (*hasTargetedLocationRelevance*) Relevance with target's location can be judged on the basis of target's and attackers mailing addresses. Cyber attack history such as sophisticated cyberattacks and APTs reveal the rivals of any victim. STIX feeds contain the threat actor's address field that identifies the location of the attacker and targeted victim's address gives the location of the target. Table 4.3 shows the victim location of famous APTs, *Red October* campaign targets the network of *KZ, Eastern Europe* and *Central Asia*, *LUCKYCAT* campaign targets *India* and *Japan*, *WildNeutron* affects the location of *GB, US, France, Russia, Switzerland, Germany* and *Austria* and *Naikon* APT victimize the surrounding of *Korea*. In Listing 4.2, relevance is derived by analyzing target's physical location. For identity characterization of incident victim and threat actor, STIX uses OASIS Customer Information Quality (CIQ) [37]. In CIQ specification field of STIX contains information regarding the address, locality, country and administrative area, used in the rule to derive location relevance with network location identified through *GeoLocation* elements.

```

STIX (?X) ∧
  hasIncident (?X, ?in) ∧
  hasVictim (?in, ?v) ∧ hasAddress (?v, ?a) ∧
  hasCountry (?a, ?c) ∧
Network (?N) ∧
  hasGeolocation_Country (?N, ?g) ∧
  swrlb:stringEqualIgnoreCase (?c, ?g) ∧
  hasAssociatedRisk (?X, ?r) ∧
  hasSTIXRelevance (?r, ?rlv)
→ hasTargetedLocationRelevance (?rlv, 1)

```

Listing 4.2: Location Relevance as SWRL Rule

Relevance with Affected Assets (*hasAssetsRelevance*) Relevance with affected assets is identified either by the type of incident as classified in STIX feed or by finding a match between the targeted asset defined in threat with the configured asset present in the network. Affected assets include PCs, Mobile phone, databases, servers, credentials and records. Table 5.3 shows the compromised assets of *Red October, LUCKYCAT, WildNeutron* and *Naikon*. *Red October* compromised the We have defined a rule that compare our defined network's assets with the instances of STIX feed in order to estimate the damage caused by particular threat in Listing 4.3.

```

STIX (?X) ∧
  hasIncident (?X, ?in) ∧
  hasAffectedAssets (?in, ?a) ∧
Network (?N) ∧
  hasNetworkAssets (?N, ?z) ∧
  swrlb:containsIgnoreCase (?z, ?a) ∧
  hasAssociatedRisk (?X, ?r) ∧
  hasSTIXRelevance (?r, ?rlv)
→ hasAssetsRelevance (?rlv, 1)

```

Listing 4.3: Assets Relevance as SWRL Rule

Relevance with Compromised CIA Property (*hasCIA_Relevance*)

Relevance with compromised CIA property depends on the organization's preference for a particular CIA property. It is derived from the incident affected element of STIX which has CIA property. We have defined a rule to identify those threats that affect or target any of CIA security property like confidentiality, integrity and availability. Table:4.3 shows that *Red October* compromise the *Confidentiality* and *Integrity* of network, *LUCKYCAT* targets the *Confidentiality*, *WildNeutron* affects the *Availability* of network and *Naikon* affects the *Confidentiality*.

Relevance with Incident Severity (*hasImpactRelevance*) Relevance with incident severity is derived for high severity threats and attacks. This severity factor is described by impact assessment element of STIX. Some common examples of impact assessment include financial loss and data breach defined by a confidence value that reflects the level of impact in terms of high, medium and Low. As an example we have defined a rule for data theft severity in Listing 4.4. The rule filters STIX feeds where confidence value is *High* and incident effected value is *DataTheft* for *Database* assets.

```

STIX (?X) ∧
  hasIncident (?X, ?i) ∧
  hasImpactAssessmentType (?i, ?t) ∧
  swrlb:stringEqualIgnoreCase (?t, "Data Theft") ∧
  hasConfidence (?i, ?c) ∧
  swrlb:stringEqualIgnoreCase (?c, "High") ∧
Network (?N) ∧
  hasNetworkAssets (?N, ?n) ∧
  swrlb:stringEqualIgnoreCase (?n, "Database") ∧
  hasAssociatedRisk (?X, ?R) ∧
  hasSTIXRelevance (?R, ?rlv) ∧
→ hasImpactRelevance (?rlv, 1)

```

Listing 4.4: Severity Relevance as SWRL Rule

Table 4.2: Observed Threat Relevance Elements

	CVE	Motiva- -tion	Victim Loca- -tion	Assets	CIA	Lang- -uage	Organiza- -tion	Impact	Secur- -ity Com- -promise
Red Octo- -ber	CVE-2008-4250, CVE-2009-3129, CVE-2010-3333, CVE-2012-0158	Espionage	KZ, Eastern Europe, Central Asia	Desktop, Mobile phone, Router, Server	Confident- -iality, In- -tegrity	English	Government, Scientific research organizations, Financial firm	high	yes
LUCK - YCAT	CVE-2010-2883, CVE-2010-3333, CVE-2010-3654, CVE-2011-0611	Political	India, Japan	Credit card, Banking information, Computers	Confident- -iality	Japanese	Military, Aerospace, Shipping, Engineering	high	yes
Wild- Neu- -tron	CVE-2012-3213	Ego, Eco- -nomic	GB, US, France, Russia, Switzerland, Germany, Austria	Web appli- -cation	Availabi- -lity	English, French	BITSTAMP, Law firms, Bitcoin-related companies, IT, Healthcare, Real estate companies	high	yes

Table 4.3: Observed Threat Relevance Elements

	CVE	Motiva- -tion	Victim Loca- -tion	Assets	CIA	Lang- -uage	Organiza- -tion	Impact	Secur- -ity Com- -promise
Naikon	CVE-2012-0158, CVE-2010-3333	Ideological	Myanmar, Vietnam, Singapore, Laos, Malaysia, and the Philippines, UN, Asia	PCs, Documents, Records, Databases, Personal details	Confident- -iality	Korean	ASEAN governmental agencies, Government departments, Investment enterprises, Military, Law enforcement, Border control organizations, Embassies	high	yes

Relevance with Language (*hasTargetedLanguageRelevance*) Relevance with language is computed on the basis of language of the victim. Some attacks target the network and people on the basis of their language, cast and nationality. We have defined rule in Listing 4.5 that detects a network’s language through the language of software installed on hosts such as operating system. This is compared with the targeted victim’s language in the STIX feed. Table:4.3 shows the the APTs that targets the victims associated with specific language. *Red October* campaign targets the *English* speaking people, *WildNeutron* victimize *French* and *English*, *Naikon* targets the *Korean* government and *LUCKYCAT* targets *Japanese*.


```

STIX (?X) ∧
  hasIncident (?X, ?i) ∧
  hasVictim (?i, ?v) ∧ hasIdentity (?v, ?id) ∧
  hasLanguage (?id, ?l) ∧
Network (?N) ∧
  hasHost (?N, ?h) ∧
  hasSoftwareInstallationLanguage (?h, ?sl) ∧
  swrlb:stringEqualIgnoreCase (?l, ?sl) ∧
  hasAssociatedRisk (?X, ?r) ∧
  hasSTIXRelevance (?r, ?rlv) ∧
  → hasTargetedLanguageRelevance (?rlv, 1)

```

Listing 4.5: Language Relevance as SWRL Rule

Relevance with Security Compromised Element (*hasSecurityCompromiseRelevance*)

Relevance with security compromised element is used to find critical threat knowledge where security is actually compromised successfully by the attacker. For this purpose, STIX contains incident security compromise element. If the value of this element is “yes”, it indicates that the attack incident is successfully accomplishment.

We have described nine major relevance computing factors denoted as F_i where $(0 \leq i \leq 9)$. If any of these mentioned elements is missing from the STIX feed, the rule will find relevance using other identified elements to map threats to the network. We can use threat attributions as well, discussed in (section 6.1) to identify the attacker’s attack pattern and the attack motivation by analyzing history or stored STIX reports found in the databases.

4.1.2 Threat Likelihood (L)

Threat Likelihood derives the score to measure the possibility of occurrences of certain nature of threats/STIX on Network. The calculation of L is based on derived relevance scores S_i as discussed in section 4.1.1. A single relevance factor presence is identified as a single unit count. If the combined relevance score $S_i \geq 1$, only then the respective STIX is considered as relevant. Each F is based on sub attributes E to derive the S_i which gives the level of similarity between the targeted threat and victim’s network. The common attributes between STIX and network $E_s \cap E_n$ instances is divided by the total available attributes \bar{E} to measure individual S_i for F . The equation 4.2 is used to calculate the relevance score. The maximum value for S_i of single F is 1.

$$S_i = \frac{E_s \cap E_n}{\bar{E}}$$

where :

E_s is set of relevance elements found in STIX

E_n is set of relevance elements received in network

\bar{E} is set of all available relevance elements

(4.2)

In order to understand the concept, consider the rule defined in Listing 4.1. It detects vulnerabilities in the network by finding a match between vulnerable software as defined in STIX feed with the actually installed network's software. To compute relevance score S_i for the vulnerability factor, the number of common vulnerable software is divided by the total software vulnerabilities present in STIX feeds. Similarly Listing 4.2, compares the locations of targeted STIX feed E_s and victim's network location elements E_n . The CIQ element of STIX used to derive location relevance comprises of many elements for identity characterization includes *Address*, *Locality*, *Country*, *Administrative area*, 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 for F_i of Locality.

Weights (W_i) have been assigned to relevance factors (F_i) by realizing the criticality and impact of F_i with respect to the network and these weights are configurable by network administrator. We have defined the weights according to following criteria: Highest weight W_i is assigned to *hasCVE_Relevance* 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 for the network. The second most highest weight W_i is assigned to affected assets relevance *hasAssetsRelevance* and CIA preference *hasCIA_Relevance* which measures the targeted and network associated quantitative and qualitative assets, respectively. STIX feed is of no use if the affected or aimed assets are not present in the network. The third highest weight is assigned to country relevance element *hasTargetedLocationRelevance*, as the sophisticated attacks are mostly targeted and aimed to affect only specific regions or people present in the targeted or neighboring locality. The identification of targeted locality is very important to analyze STIX feeds targeting network location. The motivation relevance is placed at fourth position which is important to identify the network intent related STIX feeds. The rest of the four factors are less important with the weight W_i assigned as 1. Table 4.4 depicts the relevance F with the associated

weight W_i .

Table 4.4: Relevance Factors (F) and associated Weights (W_i)

F	W_i
hasCVE_Relevance	5
hasAssetsRelevance	4
hasCIA_Relevance	4
hasTargetedLocationRelevance	3
hasMotivationRelevance	2
hasOrganizationRelevance	1
hasImpactRelevance	1
hasTargetedLanguageRelevance	1
hasSecurityCompromiseRelevance	1

The S_i is derived from Equation 4.2 and the W_i is assigned to each F_i are observed in Table 4.4. Equation 4.3 is defined to calculate L for all F based on S_i and their corresponding W_i . L is produced as summation of the product of received S_i with their corresponding W_i and is divided by the summation of the product of maximum relevance score (\bar{S}) and W_i of respective factor F_i .

$$L = \max_{0 \leq S_i \leq 1} \frac{\sum_{i=0}^N S_i \times W_i}{\sum_{i=0}^N \bar{S}_i \times W_i}$$

where :

N is number of relevance factors F

S_i is recieved relevance score for F_i

\bar{S}_i is maximum relevance score for F_i

W_i is assigned weight for F_i

(4.3)

4.1.3 Total Loss of Affected Assets (\bar{A})

After computing likelihood, the total loss of affected assets \bar{A} is calculated by analyzing both quantitative and qualitative asset values denoted as A_n and A_l respectively. Those assets in the network 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 the threat sources on the victims network, discussed in (Section 5.4.1.5). The *Quantitative Assets Loss* A_n is derived from *Affected Quantitative Assets Cost* C_n . Further C_n is divided by *Total Quantitative Assets Cost* \bar{C}_n of network that gives the value for *Quantitative Assets Loss*, stored

as A_n . On the other hand we can't measure qualitative asset loss such as data (breach, integrity and loss) statistically for the network. Therefore we define a qualitative assets level as *Scale Value V* in network ontology. It scales the quantitative assets for their qualitative values defined in terms of CIA requirements. *Scale Value V* lies in the range [0 to 100] based on the criticality of asset int terms of CIA. The product of V and C_n gives the *Affected Qualitative Assets Cost C_l* . C_l is divided by the *Total Qualitative Assets Cost \bar{C}_l* of network, to produce A_l . The imported network instances in the framework carries V for each host. If the qualitative V is undefined on network instance, the framework will assume the network asset as non-critical and will only use quantitative loss A_n as total affected loss \bar{A} . The worth, priority and preferences of network's assets for CIA is very crucial for impact I derivation. The A_l measure the worth of critical resource present in the network. Framework measures the A_n and A_l through reasoning, the *Total Loss Of Affected Assets (\bar{A})* is calculated by dividing the sum of (A_n) & (A_l) by 2 (the maximum score of (A_n) and (A_l)) for combined affected assets score, shown in Listing 4.6.

```

STIX (?X) ∧
  hasAssociatedRisk (?X, ?r) ∧
  hasQuantitativeAssetsLoss (?r, ?qn) ∧
  hasQualitativeAssetsLoss (?r, ?ql ) ∧
  swrlb:add (?a, ?qn, ?ql) ∧
  swrlb:divide (?b, ?a, 2) ∧
  → hasTotalLoss (?r, ?b)

```

Listing 4.6: Total Loss as SWRL Rule

4.1.4 Threat Reachability (R)

Threat Reachability measures the accessibility of threats and attacks infiltration to network vulnerable hosts which are directly and indirectly connected through internet. Threat reachability denoted as R determines threat impact I on network architecture by identifying the number of assets affected, exploited and exposed to a particular threat. STIX relevant threats and vulnerabilities can be ignored in the case if appropriate controls are enabled in the network, all vulnerable hosts are not exploitable on network due to reachability defined properties. Firewall isolates threats and attacks escalation through internet to victims' network. We have assumed in our *STIX-Analyzer* owl that if firewall is activated on network vulnerable hosts then the threats and attack exploits will not trigger. Through reachability, we can determine the security state for a particular section of the network. We have

assigned a score to compute reachability. This score is calculated by dividing the number of identified vulnerable hosts with total number of hosts present in the network. Reachability score value lies between 0 and 1 representing lowest and highest values for reachability. For instance, if no control is enabled and all vulnerabilities are identified, it is scored as one which means threat reachability is high. To map the concept formally we have defined rules for reachability that provides the possible paths from threat to hosts or from affected hosts to other neighboring vulnerable hosts. The rules are designed to work in steps, (a) identify vulnerable host present in the network. (b) detect reachable routers, connected with the vulnerable hosts. (c) determine the associated subnets or interconnected hosts of reachable routers. Vulnerable hosts of network are identified in Listing 4.1, the identified vulnerable host (PCs, servers) are used in Listing 4.7 to detect the exposed and reachable routers linked with the vulnerable host. Further rules are defined in *STIX-Analyzer* to measure hosts connected with reachable routers for identification of exposed subnets. The score for R is derived by dividing by number of reachable hosts with total number of available hosts.

```

Network (?N) ∧
  hasRouters (?N, ?R) ∧
  hasConnectedHost (?R, ?x) ∧
  hasVulnerableHost (?N, ?z) ∧
  swrlb:equal (?x, ?z)
  → hasReachableRouters (?N, ?R)

```

Listing 4.7: Routers Reachability as SWRL Rule

Finally to calculate threat impact, computed values of likelihood, affected asset loss and reachability are multiplied together. To elaborate the concept further, in the following section we discuss the campaign of Red October as a case study on our sampled network.

Chapter 5

Case Study

‘Case studies of failure should be made a part of the vocabulary of every engineer so that he or she can recall or recite them when something in a new design or design process is suggestive of what went wrong in the case study’

— Henry Petroski

In Chapter 5, Formulas and designed calculations to achieve results of threat likelihood, assets loss and threat reachability are described using case study. STIX of Red October campaign is analyzed on arbitrary network ontology instance to derive results that identifies the threat relevance, likelihood, assets loss, threat reachability and risk impact of Red October attack on victim’s network.

5.0.5 Case Study: Red October STIX Impact on Network

Red October campaign [38] was launched in 2012. It involved a series of attacks targeting governmental and research organization’s networks. It exploits vulnerabilities present in Microsoft Office Word and Excel. We have considered Red October [35] STIX feed as a case study to derive its risk impact on our sample network based on our proposed *STIX-Analyzer* approach. Our network model is shown in Figure 5.1. Its properties are discussed later in this section.

The analyzed Red October instance of STIX *Exploit_Target* has three major vulnerabilities associated with MS Word and Excel. The CVE id’s for

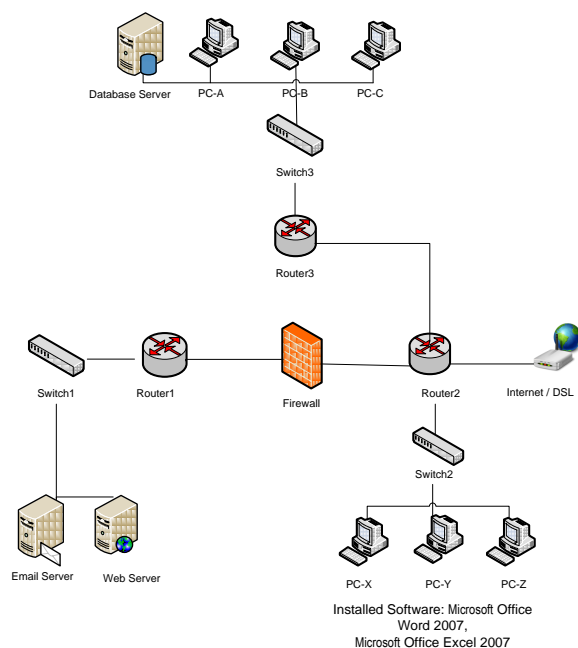


Figure 5.1: Network Example

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 different hosts in the network. As both Word and Excel vulnerabilities are present in the network to trigger the exploit, the S_i for vulnerability is derived as 1. The Red October STIX affected PCs, switches, servers, routers and cell phones. Four of these assets (PCs, switches, servers, and routers) are also present in the network, so the relevance score S_i for asset is $4/5$ i.e. 0.8. Red October STIX motivation is *finance* and targeted the US government agencies and financial organizations but the sample network instance organization type is “Health-Sector” leads to zero S_i for *motivation* and *businessstype*. The STIX targeted *country* element is “USA” and targeted *language* is “English” is compared to *GeoLocation_Country* and *Installed_Software_Language* elements present in the network instance, respectively. The matched S_i for *country* and *language* is saved as 1 but the *LocationRelevance* factor has sub attributes E_n (address, locality, administrative area) that are missing in $STIX(E_s)$, resulted in 0.25 S_i for location. The sample Health-sector network instance contains confidential data, the availability and integrity of the records are important, mentioned in the *hasCIAPreference* element of net-

work. The *hasCIAPreference* is compared with *hasAssetProperty* of red october instance, the derived S_i for *CIA* is 1, shows that the preferred CIA properties in network and targeted CIA properties of STIX are same. The relevance score computation performed by defined rules of *STIX-Analyzer* between Red October STIX and the network instance is shown in Table 5.1.

Table 5.1: Network STIX Relevance

	Red October STIX Elements (E_s)	Network Elements (E_n)	Relevance Score (S_i)
hasCVE_Relevance	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

To measure threat likelihood L ; the derived relevance score S_i is shown in table 5.1, assigned weight W_i is discussed in table 4.4 in which the maximum relevance score denoted as \bar{S} is 1 for all relevance computing factors F explained in section 4.1.1. These scores are supplied to threat likelihood formulated rule, explained using equation 4.3. Reasoning based result of threat likelihood performed by our analytics framework is shown in table 5.2. The derived likelihood score for *RedOctober* attack on observed *network* instance is 0.63.

The identified quantitative network affected assets such as PCs, servers, switches and routers with their quantitative cost C_n are shown in table 5.3. To measure qualitative cost C_l , CIA crucial qualitative network affected assets are mentioned in table 5.4. The table shows that confidential records are maintained by servers and integrity of servers with availability of servers, PC and router has assigned respective value V , from the network instance value V element. The derived sum of quantitative cost C_n is divided by the total cost \bar{C}_n (including cost of firewall 4,020.00 and internet 108) to produce total quantitative loss A_n i.e. $31083 / 35211 = 0.88$. As all the network qualitative assets are targeted, therefore qualitative cost C_l is divided by total qualitative cost \bar{C}_l to generate total qualitative loss A_l . The result

Table 5.2: Threat Likelihood (L)

F	$S_i \times W_i$	$\bar{S} \times W_i$
hasCVE_Relevance	5	5
hasAssetsRelevance	3.2	4
hasCIA_Relevance	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

derived is one. The sum of A_n and A_l is divided by 2 i.e. $1.88 / 2 = 0.94$ which gives the *total loss of affected assets* \bar{A} for network, shown in table 5.5

Table 5.3: Quantitative Asset Cost (C_n)

	Number of Affected Assets	Cost per Asset	Number of Assets \times Cost per Asset
PCs	6	\$ 1090	1090 \times 6 = 6540
Switch	3	\$ 1590	1590 \times 3 = 4770
Server	3	\$ 4593	4593 \times 3 = 13779
Routers	3	\$ 1998	1998 \times 3 = 5994
C_n			USD \$ 31083

Table 5.4: Qualitative Asset Cost (C_l)

	Critical Assets	Critical Assets Cost \times Number of Assets	V [0-100] \times Critical Assets Cost \times Number of Assets
Confidentiality	Server	13779	100 \times 13779 = 1377900
Availability	PCs, Server, Routers	6540 + 13779 + 5994 = 26313	80 \times 26313 = 2105040
Integrity	Server	13779	13779 \times 100 = 1377900
C_l			USD \$ 4860840

Threat is reachable to various nodes on network, from the internet various paths are identified to network hosts by executing the reachability rule. As the firewall is deployed between router1 and router2 which blocks the

Table 5.5: Total Loss Of Affected Assets (\bar{A})

C_n	\bar{C}_n	A_n	C_l	\bar{C}_l	A_l	\bar{A}_l
31083	35211	0.88	4860840	4860840	1	0.94

accessibility of *Red October* exploit to email and web servers of router1 from internet. The attack will infiltrate the network by exploiting the vulnerabilities of nodes directly and indirectly connected with router2 through internet. Exploit from router2 will penetrate the network by targeting the hosts of router3 which is indirectly connected to internet through router2 and targets the attached database servers which is connected with switch1. As all hosts except the hosts connected with router3 are affected the *ReachabilityScore* is $11/17 = 0.64$. The derived impact I by putting values in equation 4.1 is $0.63 \times 0.94 \times 0.64 = 0.37$.

Chapter 6

Threat Profiling

‘Cyber bullies can hide behind a mask of anonymity online, and do not need direct physical access to their victims to do unimaginable harm.’

— Anna Maria Chavez

In Chapter 6, STIX and Cybox attributes are parsed and used for threat actors profiling and threat attribution that give essence of attacks by analyzing attackers intention and motivation before particular attacker targets the victims network. The attribution is performed in three different ways, using Threat Frequency Analysis, Traffic Analysis and Threat Actors Profiling.

6.1 Proactive Detection of Threats

Threat knowledge gathered from voluminous threat repositories[33][3][34] are used for threat attribution by creating threat profiles to dynamically enforcing security perimeters for network. Threat profiling can be used by network administrator to analyse threat actors motives, goals, attack patterns, tools and methods that can be used against network. Creating defensive strategies for network and creating threat profiles are equally important to understand network critical assets, attackers interest, attack pattern followed, attackers contribution in cyber campaigns, attackers domains and address object with the traffic observed on network. Future attacks on network can be predicted by creating threat profiles and implementing adequate security controls for network.

STIX *Observables* hold various attributes and elements that provide threat profiles employed for threat attribution to their sources by analyz-

ing (i) traffic patterns, (ii) threat frequency and (iii) threat actor's profile. *CybOX* attributes for threat knowledge are useful to perform threat attribution and help in analyzing attacker's behavior, activities performed and patterns observed in network. Threat attribution is a proactive approach that provides early detection of threats before actually it is exploited. Attacker's profiles are used to reduce the anonymity of attackers. There are certain *CybOX* constructs that are used in rules defined for proactive threat detection and attribution. Major *CybOX* constructs used in threat actor's attribution are *AddressObj* for attacker's IP address and domain name, *URIObject* for malicious URLs, *EmailMessageObj* for phishing emails with attachments and file extension details, *NetworkConnectionObj* provides the protocol information with *SocketAddressObj* for socket addresses that involves IP and port (*PortObj*) information. Details regarding attacker's HTTP connection pattern is available in *HTTPSessionObj* comprises of *HTTP_Method* and *Value*. These attributes help in identifying and associating patterns followed by threat actors in conducting attacks.

6.1.1 Threat Frequency Analysis

Threat frequency analysis is used to observe threat actor's associated group, party and campaigns in which he was involved in the past. Threat frequency analysis is coupled with risk analysis discussed in section (4.1) to identify high scale and risky threats. Frequency of occurrence of a high impact threat is estimated by comparing the pattern of observed threat with the threat reports stored in STIX repositories. In Listing 6.1, *CybOX* element used for threat actor attribution i.e. *AddressObj* with recent or high impact valued STIX feed is compared with the other available instances. Then each occurrence of threat actor's domain is incremented by the frequency count and the final result are stored in the *hasAnalyzedFrequency* object.

```

STIX (NewSTIX) ∧
  hasTTPs (NewSTIX, ?T) ∧
  hasAddressObj (?T, ?y) ∧
STIX (?X) ∧
  hasTTPs (?X, ?T) ∧
  hasAddressObj (?X, ?z) ∧
  swrlb:equal (?y, ?z) ∧
  hasAnalyzedFrequency (NewSTIX, ?f) ∧
  swrlb:add (?newf, ?f, 1)
→ hasAnalyzedFrequency (NewSTIX, ?newf)

```

Listing 6.1: Frequency Analysis as SWRL Rule

6.1.2 Traffic Analysis

The frequent malicious domains with high risk impact for the network are blocked before the attacker targets the network under consideration. The malicious IPs and domain names, observed in STIX indicators and CybOX elements are blocked in network via implementing controls e.g. by enabling firewall or blocking malicious identified URIs, IPs and domain names. Our defined rule has formalized the traffic blocking mechanism by adding STIX identified malicious domains to firewall in its *hasDenyList* or in *hasBlockList*. Thus frequent and high impact *I* threat traffic from malicious sources is blocked by simply modifying firewall access state to *deny* state.

6.1.3 Threat Actors Profile

In this work, we observed that most of attack campaigns follow a pattern or use their skill set to launch a specific set of attacks. Alerts are generated by comparing the known indicators, malicious email information, malware hashes and signatures present in the indicators of STIX. An example of such an analyzed campaign is launched by *Lizard Squad*. These campaigns are observed to launch mostly DDOS attacks on the network. If this name is detected as Threat Actor's *party_name* in highly frequent and risky threats for network under consideration, then an alert for DDOS will be generated. Threat actor's *Motivation* is analysed to identify the correlation between attacker's intent and actions. Multiple STIX feeds are found where attacker's objective is to initiate an agenda or political movement that results in defacement attacks. Similarly, if *hacktivism* is detected as a threat actor's motivation, it indicates that attacker is trying to damage the reputation of the organization by propagating an agenda through defacement attack. In most of the observed STIX feeds we found that threat was from an insider who wanted to gain access to physical media and steal personal documents and credentials. Thus the type of affected asset is confidentiality. STIX where incident victims are from health or medical domains, affected assets are confidentiality and the TTP involved is data breach. Motivation behind such an attack is often related to some kind of financial gain.

Chapter 7

Evaluation

‘The first rule of any technology used in a business is that automation applied to an efficient operation will magnify the efficiency.’

— Bill Gates

This chapter 7, evaluates the quality of reasoning and contextual analysis performed by our proposed STIX-Analyzer. The proposed assessment is performed after importing XML instances of STIX, Network and CVE of various sizes. Numerous APTs, campaigns and espionages has been analyzed in terms of reasoning time, memory reservation and processor utilization using reasoning process. Formulated assessment criterion highlight the quality of reasoning performed for large networks. The proposed STIX-Analyzer framework is evaluated both structurally and conceptually to verify and validate its capability, clarity, consistency, scalability, reusability, efficiency, performance with the quality of inference performed. The evaluation process confirmed the usefulness of our proposed framework along with its compliance with the existing CTI frameworks (STIX, CybOX and CVE) and Network architecture.

7.1 Structural Evaluation

Our proposed framework comprises of ontology classes, object properties, data properties with specified domains, ranges, annotations and restrictions. Portege provides features of ontology metrics [11] and evaluation plugin [39], which are used to evaluate ontology structurally. It provide quantitative values to measure the quality of the ontology structure. The evaluation

matrices provide details regarding the hierarchy of classes; the minimum and maximum number of parents, siblings and children classes defined in the framework and the total count of object and data properties with associated domains and ranges. The count of the annotations and imposed restrictions is given on the basis existential, min, max, cardinality, hasValue and universal metrics types. Some important statistics related to ontology structure are shown in table 7.1.

Table 7.1: 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	2500
Properties with Range specified	90
Properties with Domain specified	200
Total Number of Restrictions	12263

The count for OWL entities shown in table 7.1 is fixed except the individuals and the 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. Instances are imported from three different domains (STIX, CVE, and network) and their applied restrictions type and count also varies from each other. The increase in properties and restrictions (assertions) is also observed after reasoning process, when new properties and values are derived.

7.1.1 Clarity

Our proposed framework covers the knowledge of multiple domains and is conveying the intended meaning for domain objects used. The naming conventions used for properties and concept labeling is readable and understandable. To enhance the understandability, proposed ontology for existing framework (STIX, CybOX, CVE) is using the same naming convention as specified for their domains. Using same variable names help in grasping and analyzing various concepts of different domains.

7.1.1.1 Consistency

STIX-Analyzer is logically consistent. Consistency of all the concepts, classes, relationships between properties and instances is evaluated using consistency checking through reasoner. We have chosen Pellet reasoner [12] to perform

the consistency check by evaluating the relationships between classes, sub-classes, individuals, objects, data, functional properties and restrictions. The reasoner detects and identifies incomplete and conflicting properties in the list. Using Pellet, we found no flaw and inconsistency in our proposed model as depicted in Figure 7.1. The Pellet reasoner took only few seconds to perform consistency check.

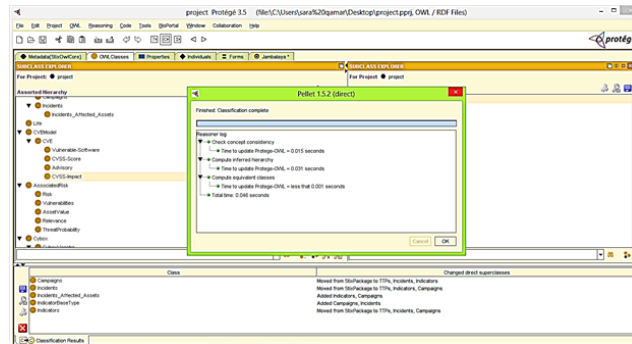


Figure 7.1: Consistency Evaluation

7.2 Conceptual Evaluation

7.2.1 Capability

The proposed *STIX-Analyzer* is following an ontological approach that has built in capabilities for contextual analysis, reasoning and inference. These features are the basic requirement to investigate extensive threat knowledge with the repositories of vulnerabilities for network entities. The developed framework is analyzing the knowledge of various domains (STIX, Network & CVE), has multiple constructs, assertions or restrictions, inferred facts with derived values through defined rules to perform threat analytics. Rule based reasoning is used to perform computations on imported individuals, associated with threat analytics framework to derive results. The reasoning process by threatand derivations of results through defined rules is illustrated in section (4.1) , (5.0.5) and (6.1) .

7.2.2 Expandability, Reusability and Scalability

Our proposed framework has imported vulnerabilities from NVD and threat knowledge from STIX repositories, but it is capable of importing knowledge from other sources as well. Ontology converts the imported XML data attributes into variables and classes based on the XML structure, if not already

defined. By renaming the variable names in defined rules (if changed) for new knowledge, reasoning and analysis can be performed. For risk analysis more factors can be incorporated by doing minor changes in defined rules. The stored threat and vulnerability knowledge can be reused for vast network architectures. New instances and knowledge can be imported or added without editing the framework. More memory is required to import huge knowledge set. The memory reservation can be minimized or maximized by doing a single line of change in the framework configuration by updating the range specified for minimum Xms and maximum Xmx memory reservation.

7.2.3 Performance

Performance of ontology is evaluated by importing number of ontology instances as discussed in section 3.1.2. The ontology instance are imported in the form of XML and the size of instances in kb is used to evaluate performance based on the attributes and elements defined in it. During performance evaluation 100kb is fixed for CVE instances and the remaining portion is divided in two halves for STIX and network instances. Various sizes of STIX is imported from the STIX repositories and is enriched from the online resources with the generation of new STIX instances from APTs reports, discussed in section (7.2.4). The size of network instance is increased by increasing the number of hosts with the information regarding the network links, vulnerabilities and software installed on hosts. Minor increase in memory reservation, inference time and processor utilization has been observed with increase in number of STIX instances and Network sizes. Major rules during reasoning process that utilizes maximum of resources in terms of time, memory and processor is discussed below.

Efficiency The efficiency of proposed framework is measured in terms of time required by rule engine to perform reasoning and analysis on a number of ontology instances. *Droolsruleengine* is used to execute *SWRL* rules to perform semantic reasoning through *Pellet* reasoner. Numerous reasoners are available to perform inference but *Pellet* takes comparatively less time in contextual analysis, inference and results derivation [25]. Reasoner took few seconds to perform inference on imported various sizes of STIX and network instances by executing defined rules of *STIX-Analyzer*. The *Droolsruleengine* measure and shows the complete inference time required by rules during execution. Figure ?? shows the relative time took by the reasoner to infer results by executing rules for *RelevanceScore(S_i)* and *ThreatLikelihood(L)*, *QuantitativeAssetsLoss(A_n)* and *QualitativeAssetsloss(A_l)*, *ThreatReachability(R)* and *ThreatActorsAttribution* from ontology individuals xml (STIX, CVE,

Network) after successful execution of rules. An increase of few seconds is observed while deriving results through inference and reasoning. Figure 7.2(a) shows the relative increase in inference time for $RelevanceScore(S_i)$ with respect to $ThreatLikelihood(L)$ as the computation performed for (S_i) is based on nine separate rules discussed in 4.1.1 and is the major rule identifying threat relevance with network. In Figure 7.2(b) it is observed that $QualitativeAssetsLoss(A_l)$ consumes time larger than $QuantitativeAssetsLoss(A_n)$ because in (A_l) all quantitative assets are calculated with respect to their assigned scale values and specified CIA preference (section 4.1.3). Similarly, the process of $ThreatActorsAttribution$ (section 6.1) is quite complex as compared to $ThreatReachability(R)$ proposed in section (4.1.4) requires more time shown in Figure 7.2(c) .

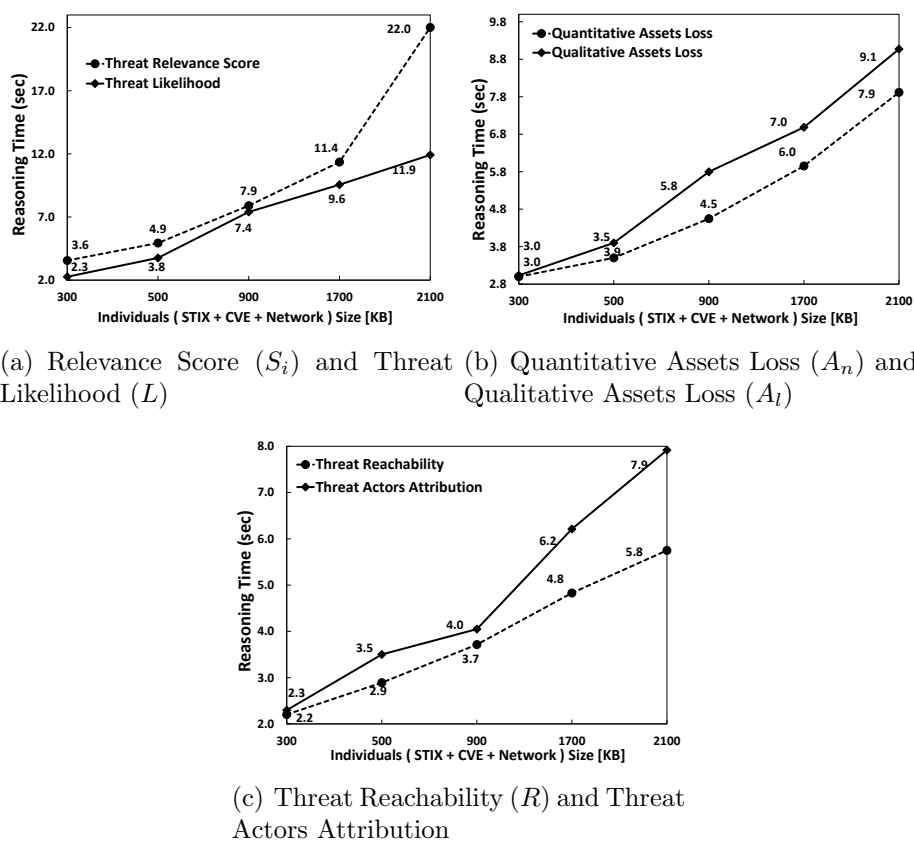
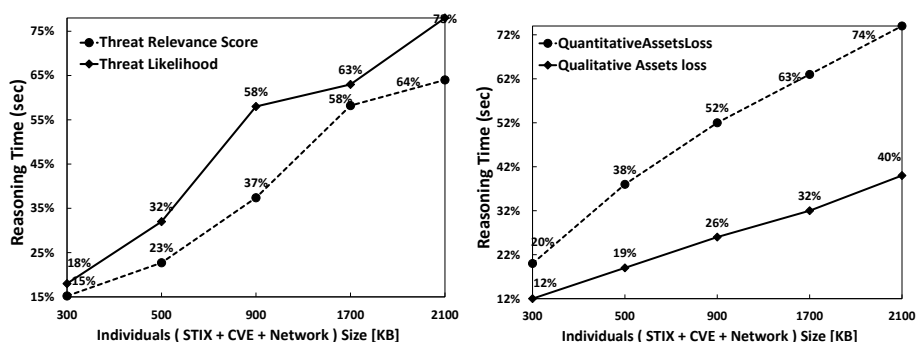


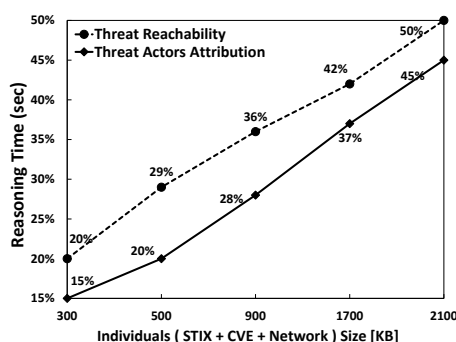
Figure 7.2: Relative inference Time (sec) during reasoning

Processor Utilization During execution of rules for inference, processor is utilized for a very short time period, as Pellet reasoner requires few milliseconds to perform reasoning on extensive STIX and huge Network instances.

The reasoning process depends on the size or the number of declared and used properties in imported instances. Figure ?? shows, the rule based relative utilization of processor while inference process, performed by *Droolsruleengine* with respect to the size of declared properties in imported instances. Similar to the inference time, usage of processor for *ThreatRelevance* computation is maximum among all defined rules.



(a) Relevance Score (S_i) and Threat Likelihood (L) (b) Quantitative Assets Loss (A_n) and Qualitative Assets loss (A_l)

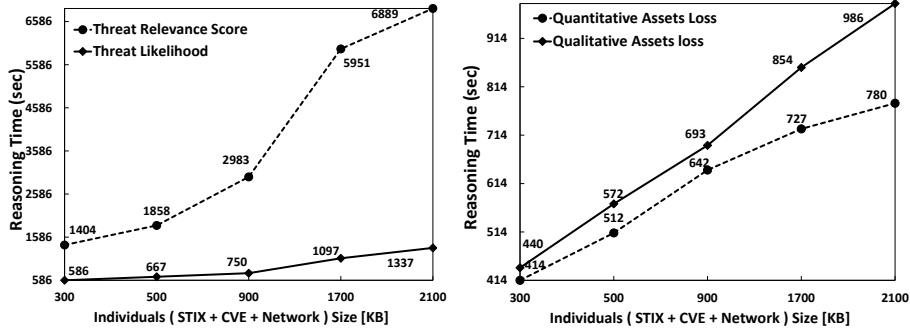


(c) Threat Reachability (R) and Threat Actors Attribution

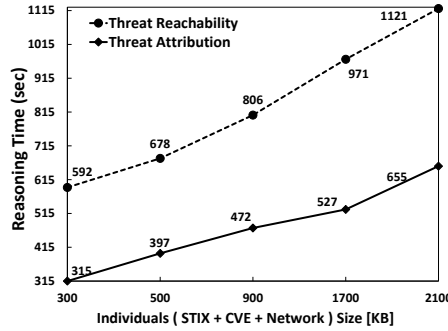
Figure 7.3: Relative CPU Utilization during reasoning

Memory Reservation All the axioms, classes, data, objects, inferred properties and instances of ontology consumes memory and the memory size is affected by both, defined and the inferred properties. Our framework is comprises of three different domains that resulted in big ontology that consumes memory when results are derived and properties are inferred. The classes, data and object properties in the framework are static and consumes fixed memory but the memory size vary from instance to instance, usage of memory increases with the number of imported instances and the inferred

properties. Figure 7.4(a) shows, relative the memory consumption by *STIX-Analyzer* during execution of major rules during reasoning process.



(a) Relevance Score (S_i) and Threat Likelihood (L) (b) Quantitative Assets Loss (A_n) and Qualitative Assets loss (A_l)



(c) Threat Reachability (R) and Threat Actors Attribution

Figure 7.4: Relative Memory reservation during reasoning

7.2.4 Reasoning Quality

Variety of APTs report and STIX have been analyzed by importing and generating instances for inference required by framework to perform threat analysis. Instances for STIX are generated by parsing the realistic APTs reports and converting into an STIX XML format used by designed *STIX-Analyzer* to perform reasoning for derivation of *Impact* and *Threat Actors Attribution*. Part of STIX XML instance is shown in Equation 3.6. After analyzing multiple STIX feeds, we found that the mentioned threats or APTs knowledge is incomplete and some of the basic attack attributes required to perform threat analytics are missing or unavailable in STIX. The missing STIX attributes were present on threats and APTs reports, so we

perform enrichment on existing STIX instances and generated new APTs XML instances to perform reasoning. In spite of missing and inaccuracy in information found in imported STIX instances, the advantage of employing different relevance computing factors F , allows rule engine to identify the relevance and threat attribution to high level of accuracy through ontology reasoning on defined rules.

Reasoning quality for relevance identification Various STIX and APT reports of different sizes has been analyzed to identify the quality of attributes present in the document required to perform reasoning. Thirty of the famous APTs and STIX report are considered below to analyse the quality of relevance attributes (*CVE, Motivation, Location, Assets, CIA, Language, Organization, Impact, SecurityCompromise*, discussed in section 4.1.1) found in their imported instances. Few STIX like Red October [31], Mandiant APT1 [40], and FireEye report on Poison Ivy [41] have been expressed in detail, but most of the remaining STIX documents that we analyzed were incomplete. Figure 7.5 represent the quality of STIX with reference to the presence of relevance factors found in APT reports. All attributes required for relevance factors identification were present in APT reports of LUCK-YCAT [42], Naikon [43], APT1 [40], Poison Ivy [41] and eight out of nine relevance factors are identified in Operation Aurora [44], Cyber attack on department of revenue [45], IXESHE [46]. Seven attributes are listed in APTs for WildNeutron [47], Miniduke [48], Red October [31], Shamoon attack [49], Italian hacking team [50], Sony entertainment breach [51], Operation Troy [52], USPS hacked via vpn connection [53], Operation Shady Rat [54], Operation Tropic Trooper [55], Ebay & Pay pal breach [56] and Exiled-Tibetan Government Website hacked [57]. Six of the listed attributes are observed in APTs regarding Heartbleed attack [58], Bit9 Inc. [59], POS Malware [60], AshleyMadison hacked [61], KASPERSKY Lab breach [62], WaterHole attack [63], Belgian telecommunication breach [64] and US data breach [65]. Four of the relevance identification attributes were observed in USPS Fraud (Man killed) [66] and Germanys police system hacked [66]. The least *Relevance Factors (F)* are found in STIX is related to FBI Investigation where Multiple banks were compromised [67], the resulted relevance score for this particular STIX is 3 (> 1) and the framework will consider this STIX relevant to perform threat analytics according to network design the framework will mark APTs or STIX irrelevant, if the found relevance score or attribute (< 1).

Reasoning quality for threat actors attribution For threat attribution quality assessment, various constructs (section 6.1) for threat actor identification are analyzed in imported instances of STIX and APTs report. After

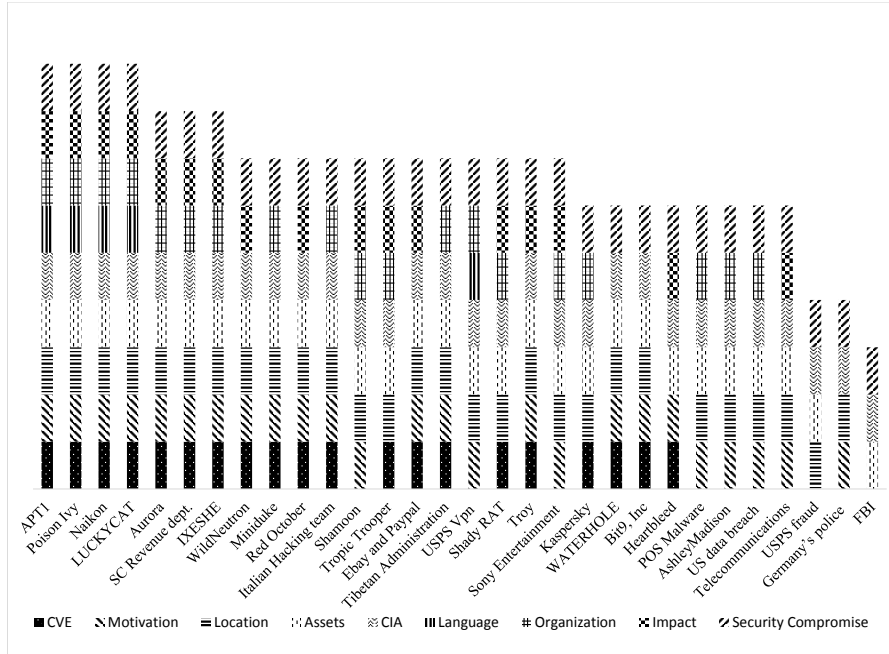


Figure 7.5: Relevance Factors (F) found in STIX

analyzing multiple STIX feeds, we found that most of the elements required to maintain threat actor's profile were missing or unavailable in STIX but are present on threats and APTs reports, so we perform enrichment on existing STIX and generated new APTs XML instances to perform *proactive threat actors attribution* using *STIX-Analyzer*. Figure 7.6 shows the observed results of thirty STIX feeds and APTs with their associated attribution elements (threat actors *Name*, *Country*, *Motivation*, *Campaign – title*, *AdministrativeArea*, *EmailMessageObj*, *AddressObject*, *OrganizationType*, *NetworkConnectionObject*, *URIObject*, *Language*, *HTTPSessionObj*, *IP* and *Type*) used by the framework. The CIQ element that represents threat actor identity was present in most of the STIX. Thirteen out of fourteen attributes for threat actors attribution were found in LUCKYCAT campaign, twelve attributes were observed in Red October campaign and APT1 report, ten elements were identified in Operation Troy, Nine threat actors profile elements were found in APTs related to IXESHE, Poison Ivy, Operation Tropic Trooper, Naikon APT, Ebay and Pay Pal breach, eight attributes were found in Operation Aurora. Few analysed APTs and attacks has less information

related to attacker, as only threat actors *Motivation* and *Type* is detected in hack of Germanys police system, POS Malware, breach of Sony online entertainment, FBI investigation because it's not appreciated to publish the names of attackers or country, industries assume it as a confidential business matter or it affects the companys reputation.

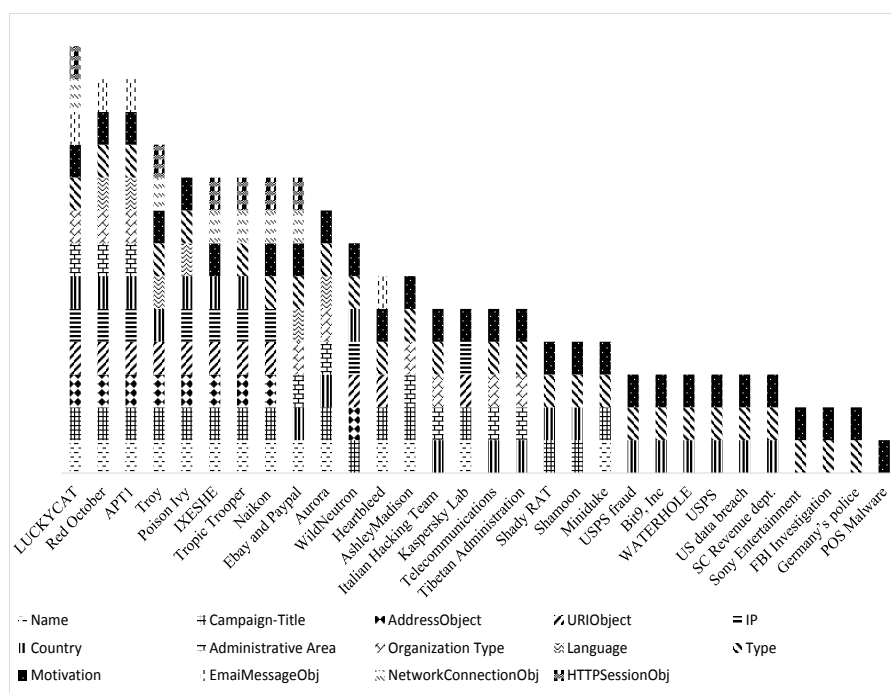


Figure 7.6: Threat Actor's Attributes found in STIX

Reasoning Quality for Impact Derivation on Networks: We have evaluated the impact I of multiple real time STIX on various networks. The repository of generated STIX and network instances are places on-line [?], few of them are shown in Fig. ???. The impact of real time STIX Red October [31], Luckycat [42], Wild Neutron [47] and Data Breached [?] is derived on the same size of network of various types, including banks, military, government, health and research institutions. The results shows that the highest impact of Red October is seen on scientific research organization, a significant impact of Luckycat is observed on military network, the impact of Wild Neutron campaign is high on health and banking sector and the Data breach attack equally targets the banks and government organizational networks.

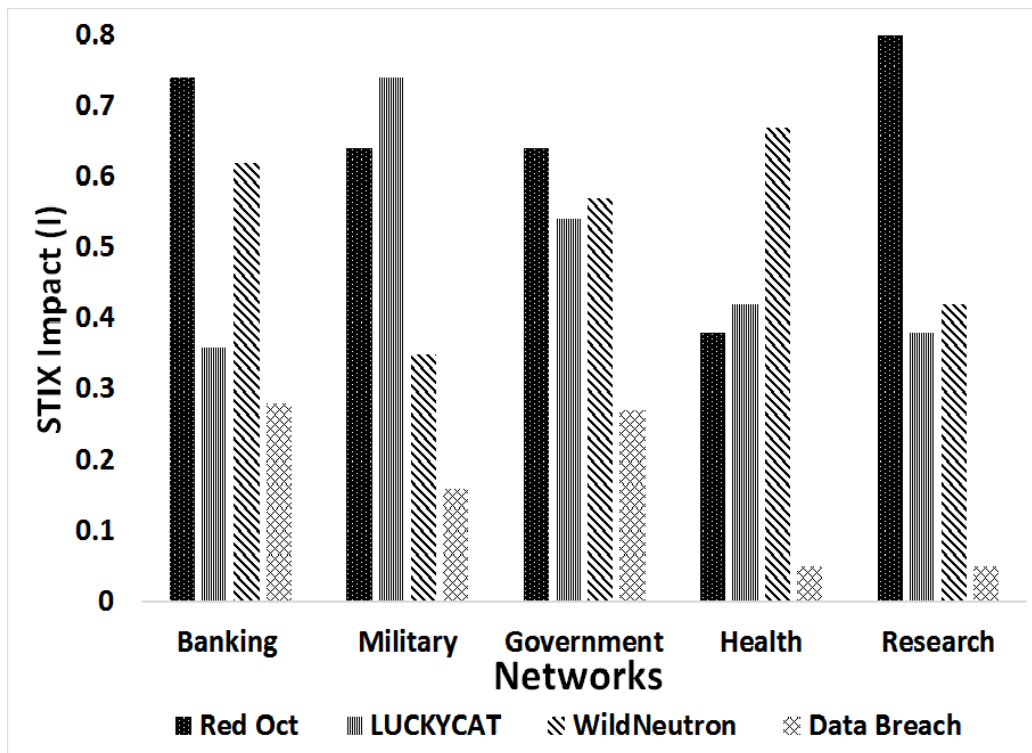


Figure 7.7: Relevance Factors (F) found in Networks

Chapter 8

Conclusion and Future Directions

‘Reasoning draws a conclusion, but does not make the conclusion certain, unless the mind discovers it by the path of experience.’

— Roger Bacon

This chapter 8 concludes the research work of threat analytics and presents the summary of thesis contribution performed to analyze threats and risk impact on network. This chapter also provides the future directions of research in threat intelligence.

8.1 Conclusion

Threat analytics provide insight of threats posed to a network in order to measure its impact and associated risks. As threats change over time with their impact on the network, we have proposed a formal framework based on ontologies that analyzes real-time threat feeds to compute relevance with network under consideration. The proposed solution measures the network associated impact I by measuring four T’s of threats. Rules are defined to analyze and investigate the information present in the STIX feeds and concluded results are mapped according to the organizations network architecture. To defend network against attacks, Proactive threat detection is discussed for network against threats by attributing threat actor’s profile as observed in STIX.

8.2 Future Work

In the future, we would like to analyze CybOX threat patterns for cyber defense in order to generate rules to perform the STIX Susceptibility Assessment [68] for a network. Enhanced threat features and more detailed research is required to design a complete strategy for proactive defense in an automated manner. More detailed information regarding threat attribution is required to completely automate the defensive mechanism for network against vulnerabilities, exploits and attacks.

Chapter 9

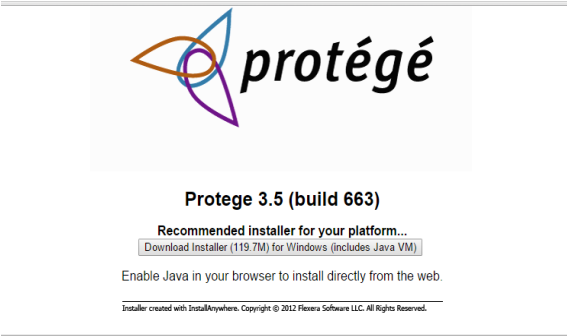
Appendix A : Setting up STIX-Analyzer

Preface

This guide elaborate the configuration procedures of STIX-Analyzer with the installation instructions for its dependencies on windows operating system. It aims to guide cyber analysts, experts and network administrators for analysis of emerging threats on network to identify threat relevance, impact derivation and for maintaining threat profiles through STIX-Analyzer deployment.

9.1 Introduction

This manual explains the configuration and installation procedures of STIX-Analyzer for windows operating system using Protege 3.5. It targets developers, cyber analysts, security experts and network administrators to comprehend and analyze voluminous threat reports, STIX feeds and attack vectors on numerous complex networks. This guide also helps the organizations to deploy STIX-Analyzer for implementing adequate controls on network to defend against critical relevant threats and attacks. The STIX-Analyzer manual provides support for network administrators to identify network critical assets, direct and indirectly reachable hosts by exploits and measures the impact. The framework aids cyber analysts to understand attacker's profiles, threat campaigns, attack patterns, methodology followed in attacks and tools used in various espionages and APTs. STIX-Analyzer and its required dependencies are open-source and freely available online, familiarity with XML is assumed for deployment. The framework is reusable, scalable and efficiently analyze huge threat reports and APTs for various industrial



Protege 3.5 (build 663)

Recommended installer for your platform...

[Download Installer \(119.7M\) for Windows \(includes Java VM\)](#)

Enable Java in your browser to install directly from the web.

Installer created with InstallAnywhere. Copyright © 2012 Flexera Software LLC. All Rights Reserved.

Available Installers

Platform	includes Java VM	without Java VM	Instructions
X MacOSX		Download (88.8M)	View
> Windows 64bit	Download (111.6M)	Download (89.1M)	View
> Windows	Download (119.7M)	Download (89.1M)	View
Linux 64bit	Download (132.9M)	Download (88.9M)	View
Linux	Download (125.7M)	Download (88.9M)	View
UNIX Any Unix Platform		Download (88.9M)	View

Figure 9.1: Protege 3.5 downloads

and organizational network.

9.1.1 Configuration

This guide elaborate the configuration steps, required for framework deployment. Protg, Pellet reasoner, Swrl, Sqwrl, XML Tab are the key components and dependencies of STIX-Analyzer for successful installations in workspace. The users must have the basic understanding of XML and software debugging capability.

9.1.2 Installing Required Software

Download *Protege 3.5* installer for windows operating system which includes Java VM for different platforms from the following URL.

http://protege.stanford.edu/download/protege/3.5/installanywhere/Web_Installers/

Figure9.1 shows the download page of Protege 3.5 web installer. After complete installation, open Protege 3.5 from installed programs, Protege.bat file will execute and install all required pluggins in the background. Protege 3.5 installation provide support for *Pellet* [69] reasoner and includes its required *Drools* rule engine [13].

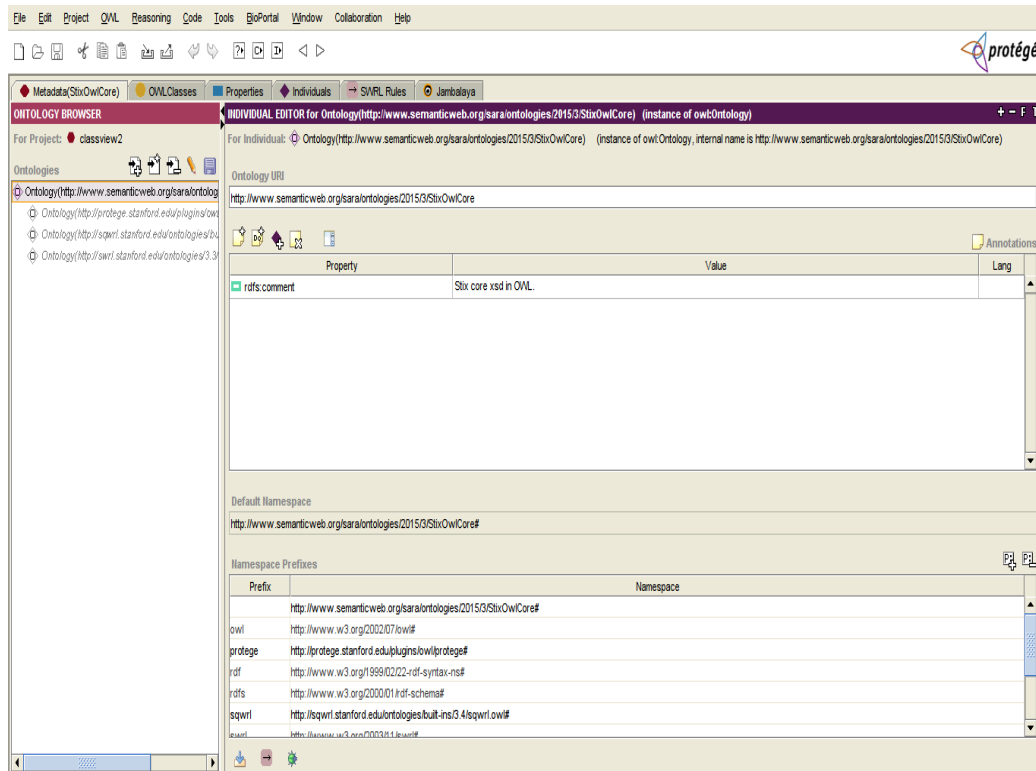


Figure 9.2: STIX-Analyzer Screen

9.1.3 Installing STIX-Analyzer

Download STIX-Analyzer from the provided URL.

<http://srg.seecs.nust.edu.pk/newsite/images/analytics/treatanalyticsframework.zip>

Unzip to extract it's owl file and open the STIX-Analyzer owl in Protege using *Open* tab from the *File* menu. Protege will load STIX-Analyzer framework with its configurations, Figure9.2 shows the Protege after loading STIX-Analyzer.

Following Tabs must be configured to view STIX-Analyzer metadata, classes, properties, instances and rules :

1. Owl Metadata.
2. Owl Classes.
3. Owl Properties.
4. Owl Individuals.
5. XML Tabs.
6. SWRL Rules.

From Protege menu, Click *Projects* and select *Configure* option shown in Figure9.3. It will open a pop up window to configure tabs. Figure9.4 shows

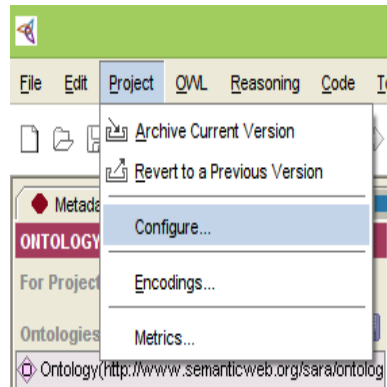


Figure 9.3: Configure options

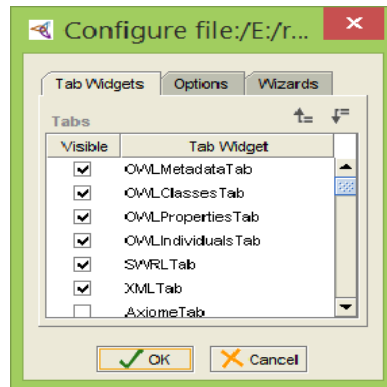


Figure 9.4: Select Configuration Tabs

the list of tabs that must be checked.

9.1.3.1 OWL Classes

Protege tab *OWLClasses* provide a complete list of defined classes and sub-classes, shown in Figure9.5.

9.1.3.2 OWL Object Properties

Protege tab *OWL Properties* gives a complete list of defined object properties and their child properties, shown in Figure9.6.

9.1.3.3 OWL Data Properties

Protege tab *OWLProperties* gives a complete list of defined data properties and associated child properties, shown in Figure9.7.

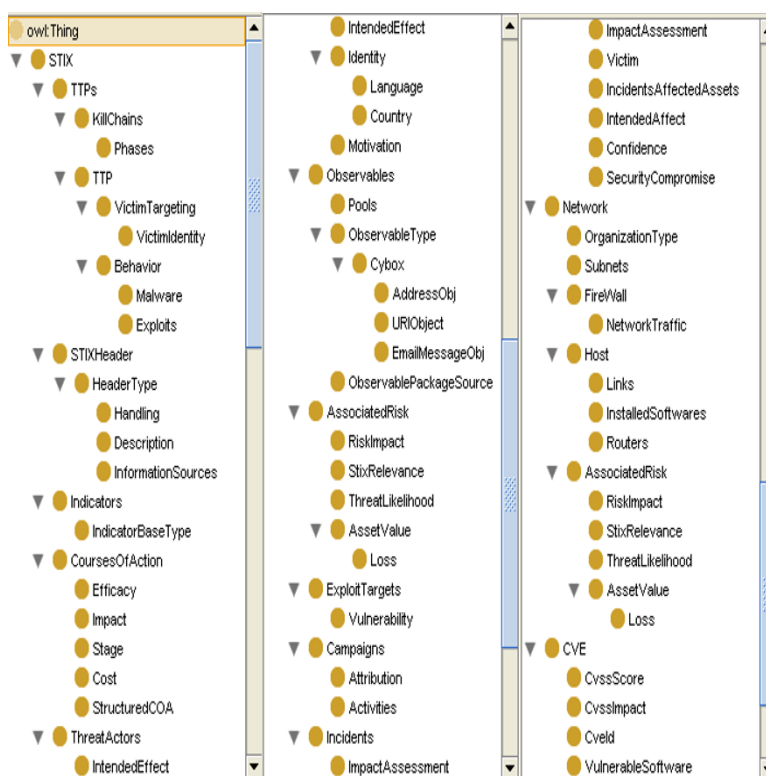


Figure 9.5: OWL Classes View

9.1.4 Importing STIX-Analyzer Instances

Analytics is performed on real time shared STIX and network models of various organizations. STIX, Network and CVE instances are imported from multiple sources to derive the impact of shared threats and STIX on network models.

9.1.4.1 STIX

STIX-Analyzer is used to detect relevant threats and possible attacks on network. For threat detection, it imports real time threat feeds from shared repositories maintained by hailataxii [4], Soltra, FS-ISAC [3] and STIXProject [34]. After minor cleaning STIX feeds are used to enrich and populate instances. The basic vocabulary of imported STIX/threat instances is provided in STIX designed template, available at URL.

<http://srg.seecs.nust.edu.pk/newsite/images/analytics/stix-template.xml>

Sample instances of famous APTs and espionage can be downloaded from the following URI.

http://srg.seecs.nust.edu.pk/newsite/images/analytics/stix_instances.zip STIX_instances.zip.

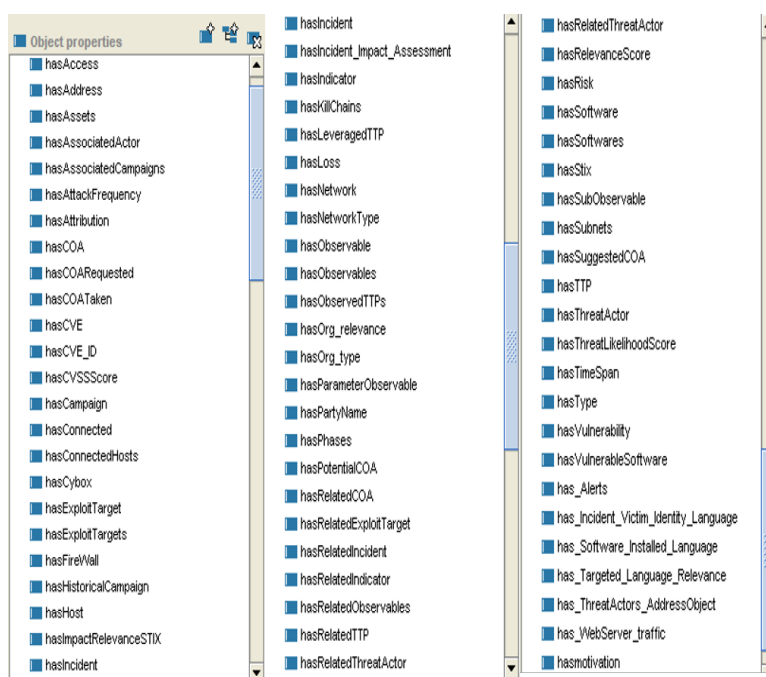


Figure 9.6: OWL Object Properties

Famous interesting APTs report are placed at the following URI <http://srg.seecs.nust.edu.pk/newsite/images/APTs>.

9.1.4.2 Network

For relevance and impact identification of threats and attacks on organization's network, network topologies are generated from BRITE topology generator and some network instances are imported from Libvirt sources. Network instance template can be download from the following URL.

<http://srg.seecs.nust.edu.pk/newsite/images/analytics/network-template2.xml>

that represents the structured network knowledge required to analyze threats on network architectures through execution of designed rules. Sample network instances of various sizes can be downloaded from

<http://srg.seecs.nust.edu.pk/newsite/images/analytics/network.instances.zip>.

9.1.4.3 CVE

CVE instances are used for network vulnerability detection and attackers exploit targets identification. Information regarding CVEs are imported from NVD repositories into ontology with complete vulnerability information. To generate instances for CVE, populate the following instance templates CVE.xml where necessary details required for vulnerability identifica-

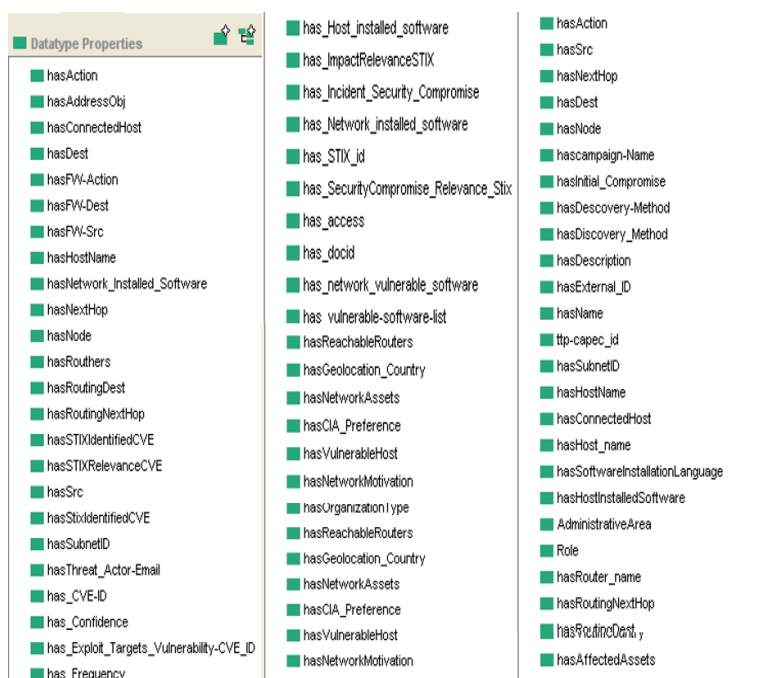


Figure 9.7: OWL Data Properties

tion are listed. Sample CVE records exploited in major attack campaigns are available at <http://srg.seecs.nust.edu.pk/newsite/images/analytics/cve-instances.zip>.

To import instances: 1. Unzip the downloaded instances files.
 2. From XML tab, Click on import button and select your desired instances (XML files) to import, as shown in Figure9.8.
 3. A pop up window of successfully imported files will be appeared.
 4. The imported instances will be populated in STIX-Analyzer Protg *Individuals* tab with their associated attributes and properties, as shown in Figure9.9.

9.1.4.4 Reasoning

Execute the defined swrl rules in sequence using *SWRL* tab and *Drools* rule engine. The values of instances data properties will be populated and updated with the inferred values and properties after reasoning process.

The Figure9.10 shows the reasoning process that involves the following steps: 1. Open Swrl tab.

2. Click on the Drools button to perform reasoning.
3. Click OWL+SWRL→Drools, it will forward the instances properties val-

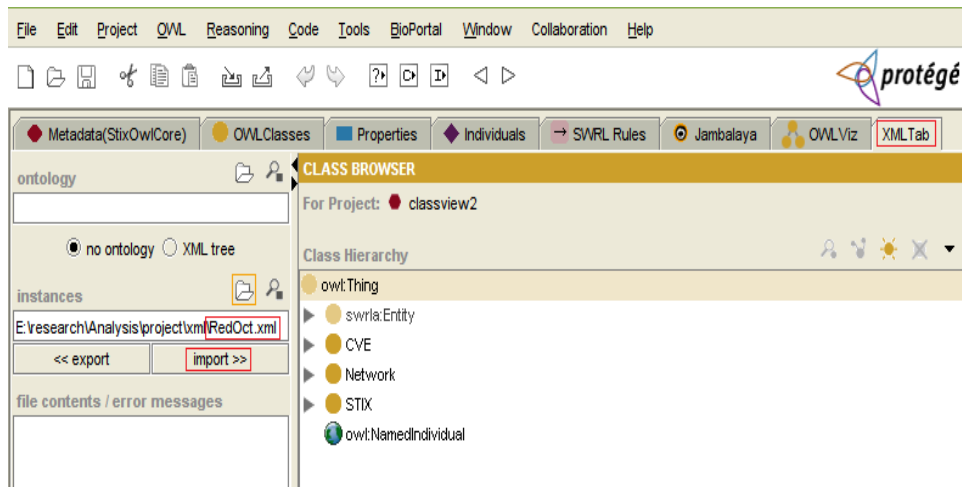


Figure 9.8: Import STIX-Analyzer Instances

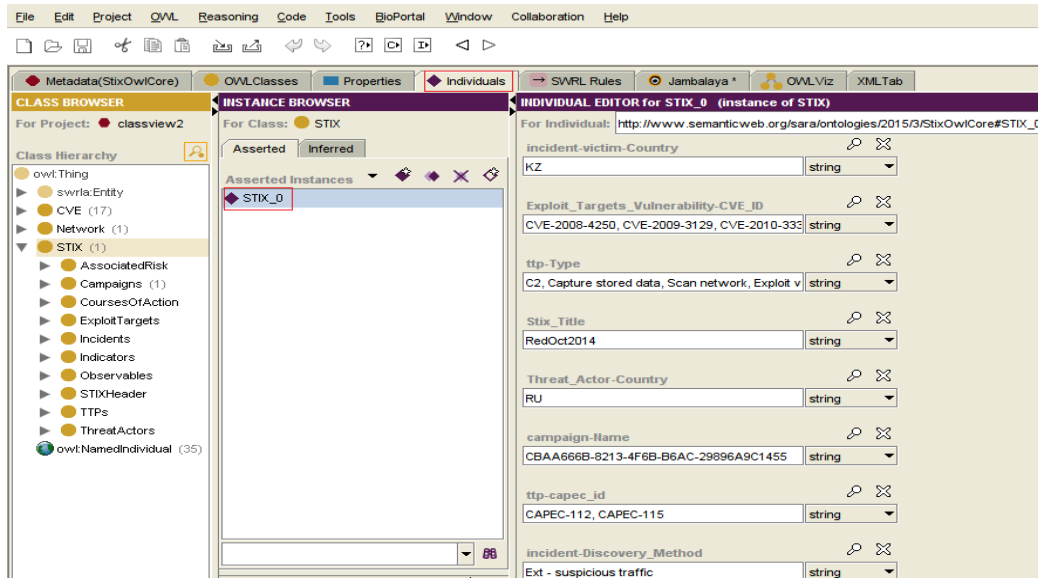


Figure 9.9: Imported Instance Attributes in Individuals Tab

ues to drools rule engine.

4. Click Run→Drools, it will perform reasoning and generate inferred values.
5. Click Drools→OWL, it will populate inferred values to instances.

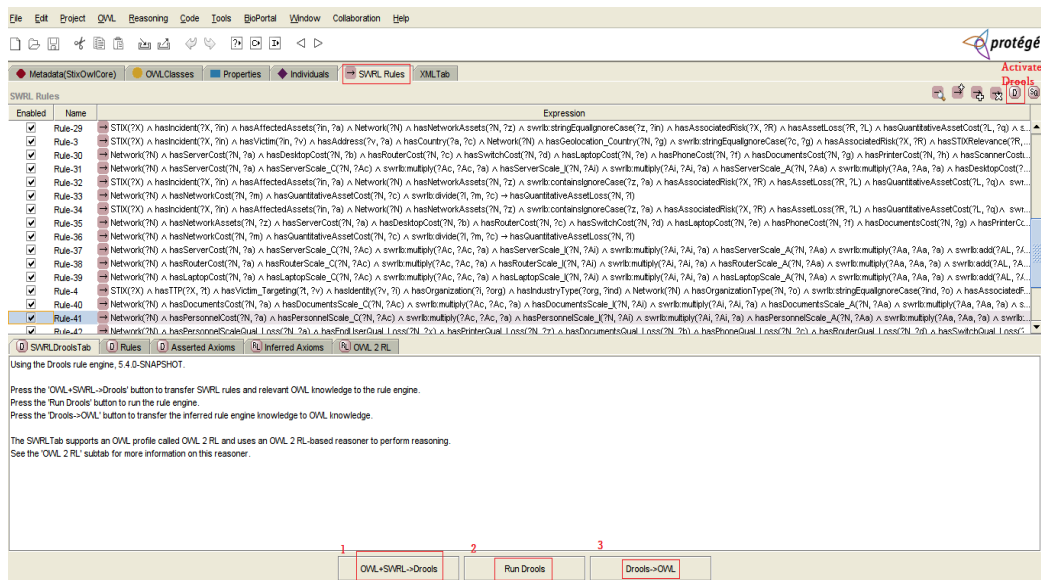


Figure 9.10: Imported Instance Attributes in Individuals Tab

Bibliography

- [1] MITRE, “Stix -structured threat information expression.” Available: stix.mitre.org/. [Accessed: 15-Apr-2015].
- [2] MITRE, “Trusted automated exchange of indicator information.” Available: taxii.mitre.org, 2015.
- [3] “Financial services - information sharing and analysis center.” Available: www.fsisac.com/, 2015.
- [4] “Hail a taxii.” Available: hailataxii.com/, 2015.
- [5] “Owl - semantic web standards..” Available: www.w3.org. [Accessed: 2015].
- [6] D. Yang, R. Miao, H. Wu, and Y. Zhou, “Product configuration knowledge modeling using ontology web language.,” 2009.
- [7] M. Stumptner, G. E. Friedrich, and A. Haselbock, “Generative constraint-based configuration of large technical systems.,” 2009.
- [8] “W3.org.” Available: www.w3.org/Submission/SWRL/. [Accessed: 2015].
- [9] MITRE, “Common attack pattern enumeration and classification.” Available: capec.mitre.org, 2015.
- [10] MITRE, “Malware attribute enumeration and characterization.” Available: maec.mitre.org/, 2015.
- [11] “Protege..” Available: protege.stanford.edu/. [Accessed: 2015].
- [12] “Complexible/pellet,.” Available: github.com/complexible/pellet, 2015.

-
- [13] GitHub, “protegeproject/swrlapi-drools-engine.” Available: github.com/protegeproject/swrlapi-drools-engine/wiki/SWRLDroolsTab, 2015.
- [14] Wikipedia, “Sparql.” Available: en.wikipedia.org/wiki/SPARQL, 2015.
- [15] F. S. Tsai and K. L. Chan, “Detecting cyber security threats in weblogs using probabilistic models,” in *Pacific-Asia Workshop on Intelligence and Security Informatics (PAISI 2007)*, Springer LNCS, vol. 4430, pp. 46-57, 2007.
- [16] Z. tang Li, J. Lei, L. Wang, D. Li, and Y. ming Ma, “Towards identifying true threat from network security data,” in *Pacific-Asia Workshop on Intelligence and Security Informatics (PAISI 2007)*, Springer LNCS, vol. 4430, pp.160 -171, 2007.
- [17] ThreatConnect, “Guide to threat intelligence platforms.” Available: go.threatconnect.com/guide-to-threat-intelligence-platform. [Accessed: 2-May-2015].
- [18] ThreatConnect, “Threatconnect and intelligence partners.” Available: www.threatconnect.com/partners. [Accessed: 1-May-2015].
- [19] J. Chavara, “Threatconnect and threat intelligence.” Available: www.threatconnect.com/news/how-to-pivoting-exporting-data-diamond-model/. [Accessed: 15-Feb-2015].
- [20] Threatstream, “Security intelligence and information sharing strategy,” Apr. 2015. threatstream.com.
- [21] ThreatQuotient, “threatq.” Available: www.threatq.com/, May 2015.
- [22] CISCO Systems, *Making Threat Intelligence Actionable: Recommending Responses with STIX*. 2015.
- [23] F. Fransen, A. Smulders, and R. Kerkdijk, “Cyber security information exchange to gain insight into the effects of cyber threats and incidents,” 2015.
- [24] E. W. Burger, M. D. Goodman, P. Kampanakis, and K. A. Zhu, “Taxonomy model for cyber threat intelligence information,” 2013.

- [25] B. E. Ulicny, "Inference and ontologies," in *Cyber Defense and Situational Awareness*, Springer.
- [26] "Visitology." Available: www.visitology.com/, 2014.
- [27] "Stix language: Version 1.2." (Archive) Available: stix.mitre.org/language/version1.2/, 2015.
- [28] NVD, "Cve_feed." Available: nvd.nist.gov/download.cfm#CVE_FEED, 2015.
- [29] protegewiki, "Owlviz." Available: protegewiki.stanford.edu/wiki/OWLViz, 2015.
- [30] protegewiki, "Jambalaya." Available: protegewiki.stanford.edu/wiki/Jambalaya, 2015.
- [31] "Github." Available: github.com/STIXProject/schemas-test/blob/a02eb29b5f655467d737f93b1bbab557083e484c/veris/4F797501-69F4-4414-BE75-B50EDCF93D6B.xml, 2015.
- [32] NVD. Available: nvd.nist.gov/download.cfm, 2015.
- [33] TAXII, "taxii-discovery-service." Available: hailataxii.com/taxii-discovery-service, 2015.
- [34] "Stixproject/schemas-test." Available: github.com/STIXProject/schemas-test/tree/master/veris, 2015.
- [35] "Nsnam." Available: www.nsnam.org, 2015.
- [36] libvirt, "Network xml format." Available: libvirt.org, 2015.
- [37] Github, "Identifying a threat actor profile." Available: stixproject.github.io/documentation/idioms/identity-group/, 2015.
- [38] GReAT, "red-october-diplomatic-cyber-attacks-investigation," 2013. securelist.com/analysis/publications/36740/red-october-diplomatic-cyber-attacks-investigation/.
- [39] protegewiki, "Ontology_evaluation." Available: protegewiki.stanford.edu/wiki/Ontology_Evaluation, 2015.
- [40] "Mandiant apt1 report." Available: stix.mitre.org/language/version1.0.1/samples/poison_ivy-stix.zip, 2015.

-
- [41] “Fireeye poison ivy report.” Available: stixproject.github.io/examples/poison_ivy-stix-1.2.zip, 2015.
- [42] “Luckycat apt.” Available: www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp_luckycat_redux.pdf, 2012.
- [43] “The msnmm campaigns.” Available: securelist.com/files/2015/05/TheNaikonAPT-MsnMM1.pdf, 2015.
- [44] “Operation aurora.” Available: kc.mcafee.com/resources/sites/MCAFEE/content/live/CORP_KNOWLEDGEBASE/67000/KB67957/en_US/CombatingThreats-OperationAurora.pdf, 2010.
- [45] “Breach in south carolina department of revenue.” Available: github.com/STIXProject/schemas-test/blob/master/veris/0DF5FC98-ADDB-48BF-8104-D20188692336.xml, 2013.
- [46] “Ixeshe apt.” Available: www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp_ixeshe.pdf, 2012.
- [47] GReAT, “Wild neutron economic espionage threat actor returns with new tricks.” Available: securelist.com/blog/research/71275/wild-neutron-economic-espionage-threat-actor-returns-with-new-tricks/, 2015.
- [48] GitHub, “Miniduke.” Available: github.com/STIXProject/schemas-test/blob/master/veris/22797F34-F191-4006-89D3-02B8ACF1555B.xml, 2015.
- [49] “Shamoon.” Available: github.com/STIXProject/schemas-test/blob/master/veris/1F0BE45F-3538-4B89-9684-C3DF47E96BD1.xml, 2012.
- [50] “Breach of italian pentesting company.” Available: drive.google.com/file/d/0Bw35r_AUuldgMEZUdXUyWUR0T3M/view?pli=1, 2014.
- [51] “Sony online entertainment : Millions of personal and credit card information and personal details were stolen..” Available: resources.infosecinstitute.com/cyber-attack-sony-pictures-much-data-breach/, 2011.

- [52] K. Alintanahin, “Dissecting operation troy.” Available: www.mcafee.com/sg/resources/white-papers/wp-dissecting-operation-troy.pdf, 2013.
- [53] “Usps hacked.” Available: abcnews.go.com, 2015.
- [54] R. Sherstobitoff and I. Liba, “Revealed: Operation shady rat.” Available: www.mcafee.com/us/resources/white-papers/wp-operation-shady-rat.pdf, 2015.
- [55] “Operation tropic trooper.” Available: www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-operation-tropic-trooper.pdf, 2012.
- [56] “Ebay & pay pal, uk domains hacked..” Available: www.zdnet.com/article/ebay-and-paypal-uk-domains-hacked-by-syrian-electronic-army/, 2014.
- [57] “Hackers infect exiled-tibetan government website with spyware.” Available: www.redorbit.com/news/technology/1112922308/dalai-lama-government-website-hacked-infected-spyware-081313/, 2015.
- [58] “Heartbleed attack.” Available: zmap.io/heartbleed/, 2014.
- [59] “Bit9 breach.” Available: krebsonsecurity.com/2013/02/bit9-breach-began-in-july-2012/, 2012.
- [60] J. Yaneza, “Malumpos.” Available: documents.trendmicro.com/images/tex/pdf/MalumPOS%20Technical%20Brief.pdf, 2015.
- [61] krebsonsecurity, “Online cheating site ashleymadison hacked.” Available: krebsonsecurity.com/2015/07/online-cheating-site-ashleymadison-hacked/, 2015.
- [62] KASPERSKY, “The duqu 2.0.” Available: securelist.com/files/2015/06/The_Mystery_of_Duqu_2_0_a_sophisticated_cyberespionage_actor_returns.pdf, 2015.
- [63] “Watering hole..” Available: www.symantec.com/content/en/us/about/media/pdfs/b-istr_18_watering_hole_edits.en-us.pdf, 2012.

- [64] “British hacked into belgian telecommunication company.” Available: theintercept.com/2014/12/13/belgacom-hack-gchq-inside-story/, 2015.
- [65] “Us nation wide data breach (jimmy johns).” Available: www.jsonline.com, 2015.
- [66] “United states postal service fraud (man killed).” Available: github.com/STIXProject/schemas-test/blob/master/veris/25EE0CC5-C946-4291-B4ED-2F6EC48334AD.xml, 2010.
- [67] “Notes: Large, ongoing incident. fbi investigation. multiple banks..” Available: dealbook.nytimes.com, 2014.
- [68] MITRE, “Cyber threat susceptibility assessment,” September 2013. www.mitre.org/publications/systems-engineering-guide/enterprise-engineering/systems-engineering-for-mission-assurance/cyber-threat-susceptibility-assessment.
- [69] H. Knublauch, R. W. Fergerson, N. F. Noy, and M. A. Musen, “The protg owl plugin: An open development environment for semantic web applications,” in *The Semantic Web ISWC 2004*, Springer.
- [70] US-CERT, “Information sharing specifications for cybersecurity.” Available: www.us-cert.gov/. [Accessed: 1-May-2015].
- [71] Prnewswire, “Prnewswire news releases.” Available: www.prnewswire.com. [Accessed: 24-Apr-2015].
- [72] M. N. Alsaleh and E. Al-Shaer, “Enterprise risk assessment based on compliance reports and vulnerability scoring systems,” in *Proceedings Of The 2014 Workshop On Cyber Security Analytics, Intelligence And Automation*, 2014.
- [73] “Stix -samples.” Available: mitre.org/language/version1.0.1/samples.html, 2015.
- [74] MITRE, “Cyber observable expression.” Available: cybox.mitre.org/, 2015.
- [75] E. H. et al., “Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains,” in *Proc.*

6th Int'l Conf. Information Warfare and Security (ICIW 11), Academic Conferences Ltd., 2010.

- [76] N. Moran and B. Koehl, "Nuzzel - htexploitteleme-try." Available: drive.google.com/file/d/0Bw35r_AUULdgakVybXBpWUZSNGs/view?pli=1, 2015.
- [77] "Germanys federal police system." Available: github.com/STIXProject/schemas-test/blob/master/veris/12377004-44BE-4C1C-98FF-EC9F841B5187.xml, 2010.
- [78] "Protege 3.5.." Available: http://protege.stanford.edu/download/protege/3.5/installanywhere/Web_Installers/. [Accessed: 2015].