18th International Conference on Information Fusion
Washington, DC - July 6-9, 2015

A Situation Assessment Framework for Cyber Security Information Relevance Reasoning

Shan Lu
Department of Electrical and Computer Engineering
Northeastern University
Boston, Massachusetts 02115
shanlu@ece.neu.edu

Mieczyslaw M. Kokar
Department of Electrical and Computer Engineering
Northeastern University
Boston, Massachusetts 02115
kokar@coe.neu.edu

***Abstract*—Cyber security is one of the most serious economic and national challenges faced by nations all over the world. When a cyber security incident occurs, the critical question that security administrators are concerned about is: What has happened? Cyber situation assessment is critical to making correct and timely defense decisions by the analysts. STIX ontology, which was developed by taking advantage of existing cyber security related standards, is used to represent cyber threat information and infer important features of the cyber situation that help decision makers form their situational awareness. However, due to the widespread application of information technology, security analysts face a challenge in information overload. There are still huge volumes of low level observations captured by various sensors and network tools that need to be used to derive the high level intelligence queries such as potential courses of action and future impact. Therefore, identification of the relevant cyber threat information for a specific query is a crucial procedure for cyber situation assessment. In this paper, we leverage the STIX ontology to represent cyber threat information in a logical framework. In order to recognize specific situation types and identify the minimal and sufficient information for answering a query automatically, we propose an information relevance reasoning mechanism based on situation theory. Finally, we implement our proposed framework using a dataset generated by Skaion corporation.**

I. Introduction

Nowadays, cyber security is one of the most serious economic and national challenges faced by nations all over the world. Cyber attacks have increased dramatically over the last decade, exposing sensitive personal and business information, disrupting critical operations, and imposing high costs on the economy [1]. When a cyber security incident occurs, the critical question that security administrators are concerned about is: What has happened? To answer this question, a security analyst should have a good understanding of the up-to-date information of the cyber situation, which includes vulnerabilities, weaknesses and configurations of current network environment, the content and patterns of the attacks, and the behavior, capability and intent of the adversary. In other words, having an efficient assessment of the current cyber situation in hand is the foundation for successful cyber defense decision-making. In this paper, we will focus on a situation assessment framework for cyber defense.

The most commonly used definition of *situation awareness* was provided by Endsley in [2]: "Situation awareness is the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future." It consists of the *perception* of environmental elements, the *comprehension* of their meaning, and the *projection* of their status for the near future [3]. Endsley distinguished the term "situation awareness" as a state of knowledge, and the term "situation assessment" as the process of achieving, acquiring, or maintaining situation awareness. Although Endsley's situation awareness model was initially developed to capture the human situation awareness, it has also been applied to the representation of the awareness of computer agents. It has also been integrated into the Joint Directors of Laboratories (JDL) model for data fusion [4].

In the cyber security domain, Barford [5] gives more detailed requirements of situation awareness for cyber defense: (1) be aware of the current situation; (2) be aware of the impact of the attack; (3) be aware of how situations evolve; (4) be aware of actor (adversary) behavior; (5) be aware of why and how the current situation is caused; (6) be aware of the quality of the collected situation awareness information and the decisions derived from this information; (7) assess plausible futures of the current situation.

A number of models have been proposed for cyber situation assessment, c.f. [6][7][8][9]. Clearly, since multiple sources of information, including humans, are involved in the process, there is a need to exchange information among all such agents while insuring that the information is both human-readable and machine-understandable. By "machine-understandable" we mean that a computer agent can parse, interpret, integrate (or even fuse) it with its own knowledge, and infer how to react to such received information.

Various cyber security standards have been proposed to represent some aspects of cyber security information [10][11][12][13][14]. Structured Threat Information eXpression (STIX) [10] is one of the most comprehensive efforts to unify cyber security information sharing. It incorporates vocabulary from several other standards. STIX captures concepts of cyber threat intelligence information and provides a high level framework to hold the various cyber intelligence

components together. These include: (1) Observables and Indicators (2) Incidents (3) Tactics, techniques and procedures of attackers (TTP) (4) Exploit Targets (5) Courses Of Action (6) Campaigns and Threat Actors. An XML schema implementation of the full STIX architecture is available [10].

Although these cyber security standards provide vocabularies for representing various aspects of structured cyber threat information and improve the consistency, efficiency, interoperability, and overall situational awareness, they still do not provide sufficient support for the semantics of the vocabulary terms that would allow computer agents to infer the consequences of the information provided to them. As described in Kokar et al. [15], in order to be aware, "...one needs to have data pertinent to the objects of interest, some background knowledge that allows one to interpret the collected object data and finally a capability for drawing inferences." In other words, situation assessment requires that important features of the situation be automatically inferred based on the data collected by network sensors and host-based applications. Automatic inference, in turn, requires that information be encoded in a language with formally defined syntax and semantics. To this end, the authors in [16] have converted some of the cyber security standards expressed in XML into ontologies expressed in OWL (Web Ontology Language) [17]. This effort resulted into the STIX ontology [18]. Ontology, which is a term used in the knowledge representation domain, stands for an explicit, formal, machine-readable semantic model that facilitates knowledge sharing and reuse. It defines the classes, instances of the classes, inter-class relations and data properties relevant to a problem domain [19]. The advantage of an ontology based approach to situation awareness is that once facts about the world are stated in terms of the ontology, other facts can be inferred using an inference engine automatically. In this paper, we will use the BaseVISor inference engine [20]. OWL is the most commonly used language for expressing ontologies today.

Due to the widespread application of information technology, security analysts face the challenge of information overload, rather than the lack of information. Although STIX ontology has some constrains for inferring the cyber security indicators and other intelligence concepts, there still is a huge volume of low level observations captured by the various sensors and network tools from which the high level queries that concern potential courses of action and future impact need to be inferred. In other words, a reasoning mechanism in the comprehension step of cyber situation assessment is needed to support the human analyst. Moreover, it is important to filter out the information that is not relevant to a specific analyst's query. This is important from both the inference mechanism's point of view (not to be overwhelmed) as well as the communications point of view. When security analysts and agents share threat information and the situation they are dealing with, sending too much information over communication links with limited bandwidth is not good. Since cyber situation assessment is a time critical process, only the information that is necessary to assess a given situation should be sent.

The objective of our work is to develop an information relevance reasoning mechanism for cyber security queries in cyber situation assessment. Although there is a lot of research about cyber situation assessment, unfortunately, a comprehensive definition for "cyber situation" does not exist. Most of the existing models describe the cyber situation by a collection of threat indicators. However, the information overload problem is still unsolved since it is not clear which *facts* need to be included in the descriptions of situations. In this paper, we will treat cyber situations as "first class citizens". This means cyber situations are separate objects which can have types and properties associated with them. The cyber situation types and situation properties can be used to not only identify what the situation is, but what the minimal amount of relevant information for decision makers to sufficiently characterize a given situation and answer a given query is, as well. To this end, we will propose a formal definition of cyber situation based on Barwise's situation theory [21]. We will extend the STIX ontology by adding a portion of a situation theory ontology structure to capture the dynamic properties of situations and to emphasize the recognition of specific situation types. This cyber situation ontology will allow computer agents to identify the minimal and sufficient information for answering a query automatically.

The rest of this paper is organized as follows: Section II overviews situation theory and STO-L ontology. Section III introduces how the STIX components work in our cyber situation assessment framework. In Section IV, we discuss the details of the information relevance reasoning mechanism. Section V describes a cyber attack scenario example and the Skaion dataset. In Section VI, we implement our proposed framework on the Skaion dataset. Finally, conclusions and future work are discussed in Section VII.

II. Overview of Situation Theory and STO-L ontology

A. Situation Theory

In this paper, we use the notion of situation introduced by Barwise and Perry as a means of giving a more realistic formal semantics for speech acts than what was then available [21][22][23]. In contrast with a (complete an universal) "world" which determines the value of proposition in a more traditional approach, a situation corresponds to the limited parts of reality that we perceive, reason about, and live in. Devlin subsequently extended Barwise's situation semantics by formalizing a number of concepts developed by Barwise and Perry [24][25]. In situation theory, information about a situation is expressed in terms of *infons*. Infons are written as

$$\ll R, a_1, ..., a_n, 0/1 \gg$$

where R is an n-place relation and $a_1, ..., a_n$ are objects appropriate for R. The last item in an infon is the polarity of the infon. Its value is either 1 (if the objects stand in the relation R) or 0 (if the objects don't stand in the relation R). It should be noted that infons are different from facts in the knowledge base. Devlin states that "infons are not things that

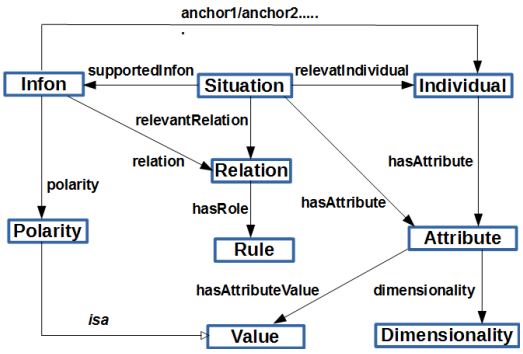

Fig. 1. Top Level Classes of STO-L

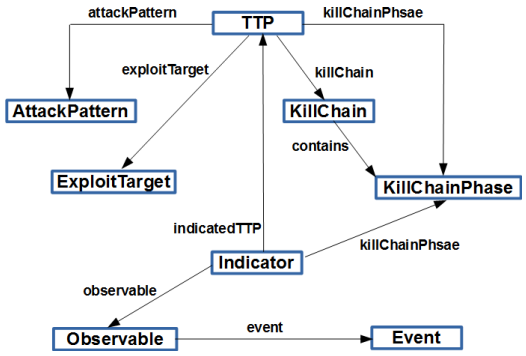

Fig. 2. Some of STIX Ontology Classes

in themselves are true or false. Rather a particular item of information may be true or false about a situation."

To capture the semantics of situations, situation theory provides a relation between situations and infons. This relationship is called the *supports* relationship which relates a situation with the infons that "are made factual" by the situation. Given an infon σ and situation s the proposition "s supports σ" is written as

$$s \models \sigma$$

In Devlin's situation semantics, a particular feature of intelligent behavior is the recognition of types. In other words, the intelligent agent recognizes objects based on their types. Since situation is an object, it could be recognized by its type. Situation theory provides various mechanisms for defining situation types.

B. STO-L Ontology

Situation Theory Ontology (STO) is a formalization of Barwise's situation semantics in terms of an ontology [15]. The basic elements of situation theory are *objects* and *types*. Each kind of object has two associated classes: the type of object class and a class that collects instances of a given type. STO is compatible with the current thinking about situation awareness in the community. In particular, there are clear relations between the concepts in this ontology and Endsley's model of human situation awareness.

STO-L is a lighter version of STO. It is a simplification and improvement of STO. Generally, there are two main modifications. First of all, we give up the meta-types of situation theory to reduce the complexity of STO. The STO-L only keeps one meta-level - the objects. Moreover, we extend the term "situation" as an entity that can affect, exhibit and participate in various actions, instead of a static collection of objects and relations among them. Figure 1 shows the top level classes of STO-L. We extend the STIX ontology into a cyber situation ontology by adding portion of STO-L structure to capture the dynamic properties of situations and to emphasize the recognition of specific situation types.

III. STIX Components in the Cyber Situation Assessment Framework

STIX ontology is created on the basis of the STIX standard [16]. It provides a common mechanism for addressing structured cyber threat information in cyber situation assessment. Some classes of the STIX ontology are shown in Figure 2. In the following, we compare the main STIX ontology classes with the concepts in Barwise's situation semantics and discuss how the STIX components work in our cyber situation assessment framework.

A. TTPs v.s. Queries

The central class in STIX ontology is *TTP* (Tactic, Techniques and Procedures of the attackers). The instances of the *TTP* class are cyber security exploits, which may belong to a variety of TTP subclasses. TTPs are related to instances of other STIX high level component classes, such as *AttackPattern*, *ExploitTarget*, *KillChain*, *KillChainPhase*, and so on.

Although both TTPs and queries basically represent the goal of situation assessment, they are different. In situation semantics, queries are outside of the situations; they are the start points of the situation assessment. In the STIX ontology, TTPs are the outcomes of the situation assessment - the awareness of the cyber situation. Therefore, the TTPs can be considered as answers to the queries from cyber security analysts.

B. Indicators v.s. Infons

The *Indicator* class is another key STIX component that plays a pivotal role in cyber situation assessment. It is a fundamental element of intelligence that connects low level observables with high level cyber intelligence concepts. The instances of *Indicator* class are patterns or behaviors that indicate the likelihood and possibly predictability of a cyber threat. Generally, an indicator could be any piece of information that objectively describes an intrusion. On the one hand, indicators are derived from observables combined with contextual information intended to represent artifacts and/or behaviors of interest within a cyber security context. On the other hand, the indicators can be potentially mapped to a related TTP context and adorned with other relevant high level

STIX components, such as kill chain. The bottom-up inference related to indicators is the key procedure in cyber situation assessment.

In situation semantics [25], infons are used as the fundamental elements to express all the information about situations. In STO-L, we only use infons to represent queries. The rest of the information about situations is expressed in OWL. The *Infon* class is the starting point of all the inferences in our situation assessment. It is also a connection between situation types and relevant individuals and relevant relations (cf. Figure 1). Infons thus play an indispensable role in our situation assessment framework because they capture the focus of attention of a situation. They allow agents to infer which facts are relevant to a given query and which are not. The inference of the *supports* relation between infons and specific situation types provides us with a rule of what is relevant for a specific situation.

C. Kill Chains v.s. Situation Types

In STIX ontology, the *KillChain* class is used to show the multiple steps in an attack. Various low level incidents are correlated as part of a larger kill chain. From low level indicators, a kill chain with kill chain phases can be inferred based on the potential threat actor and target. Therefore, the kill chain can be viewed as a particular type of situation, in which several events of particular types occur in a specified order.

In STO-L, different situation types are denoted as subclasses of *Situation*. The situation types and situation properties can be used to not only identify what situation it is, but what the necessary relevant information is to sufficiently characterize a given situation and answer a given query is as well.

In our framework, the *Event* class was used to capture the low level events data that is captured in typical log data. Events and the entities related to events are treated as individuals in STO-L.

D. Cyber Situation Assessment Framework

Figure 3 illustrates the architecture of our cyber situation assessment framework. The inputs of this framework have two parts: the queries from cyber security analysts, which may involve various aspects of cyber security intelligence information; and the voluminous observations and data about the network captured by sensors and network tools.

The outputs of this framework also contain two parts. The first part is the answers to cyber security analysts' queries. The answers could be either yes/no boolean types, or a collection of OWL facts for complex questions. The latter is the minimal and sufficient set of information for answering the queries, which can be used by decision makers to sufficiently characterize a given situation and communicate the situation descriptions to others.

The core component in this cyber situation assessment framework is the information relevance reasoning mechanism. This mechanism identifies the minimal and sufficient relevant information for each query. The details of this mechanism will be discussed in next section.

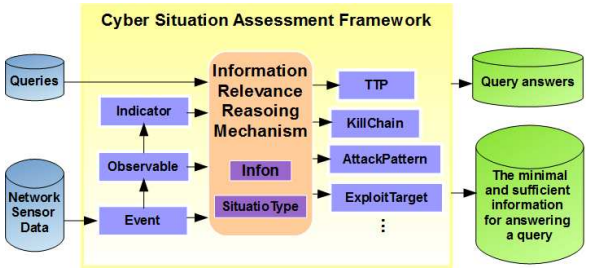

Fig. 3. Cyber Situation Assessment Framework Architecture Overview

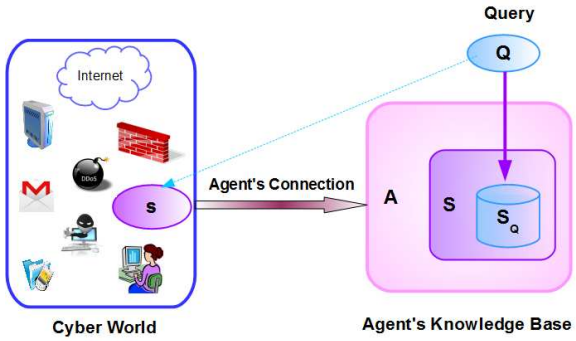

Fig. 4. Agent's Knowledge Base

IV. Cyber Security Information Relevance Reasoning Mechanism

A. Information Flow in Agent's Knowledge Base

First we relate the cyber security situation assessment to the Barwise's situation semantics and the STO-L structure. Figure 4 shows a cyber security agent's knowledge base on the right and a cyber world on the left. Cyber world means the physical (or abstract) world that is the subject of cyber security. The circle labeled s in the cyber world is a situation. This is a real situation that is happening in the cyber world. For example, this situation could be an ongoing cyber attack. Agent can be any intelligent agent who has the capability of inference, such as human or computer. Basically, that agent uses sensors or other information technologies to observe some aspects of the real world.

As we mentioned before, Barwise's situation semantics states that the relation between a situation (in the world) and a representation of the situation (in a formal framework) is relative to a specific agent who establishes such a link. This link is defined by the *Agent Connections* that link entities in the world to the formal constructs of the situation-theoretic framework. In Figure 4, the agent's knowledge about the world, that has been acquired via Agent Connections, is represented by the rectangle labeled A. In this paper, we will use the knowledge-based system approach to reason about information relevance. The idea is to store knowledge in a knowledge base which is combined with a reasoning mechanism that is used to determine what can be inferred from the facts in the knowledge base. The rectangle A is a knowledge base of an agent about

parts of the cyber world. The small rectangle S in A denotes an *abstract cyber situation*, i.e., the agent's formally represented knowledge about s.

The circle above the rectangle A represents a query, Q. The agent need a start point for a specific cyber situation assessment, which is a perspective that gives focus to what should be considered as relevant for this cyber situation (situation s for example). In STO-L, it represented as "utterance" - an instance of class *Infon*.

B. Mapping Queries to Formal Language

A query is the starting point of situation assessment and decision making processes of intelligent agents. Queries are usually represented as expressions in natural language or in a query language. In order to acquire the meaning of the query, this expression needs to be represented in a formal language. In this paper, we formalize the process of information relevance reasoning using a STO-L. So the first step is to map a query expressed as a sentence to OWL. The difference between queries to a database and queries to an OWL knowledge base is that the answer to a knowledge base query may include facts that are inferred as well as facts that have been explicitly asserted.

The query Q in Figure 4 is expressed as an instance of the class *Infon* in STO-L:

$$Q = \ll r_Q, a_1, ..., a_n, 0/1 \gg \tag{1}$$

where r_Q denotes an n-place relation in the query Q. The connection between a situation s and a collection of infons Q would be expressed using the $supports$ construct

$$s \models Q \tag{2}$$

In this conceptualization, the objective of situation assessment is to infer whether the specific situation that the analyst is querying about holds or not. This inference will be described in Section IV-D.

C. Query and Situation Type

Our final goal is to identify the minimal set of relevant information to answer a given query. So after we represent the query as an instance of the *Infon* class, the next step is to establish the information base that includes relevant information about the situation happening in the cyber world. In Figure 4, it is represented by the small rectangle S. How do we know what it contains?

The intelligent agent recognizes situations based on their types. In STO-L, different situation types are denoted as subclasses of the *Situation* class. The main purpose of reasoning about situations is to find out what situation type a specific situation s belongs to. For a given query, Q, the situation type $\bar{S}$ in situation theory is defined as (where $\dot{s}$ are instances of the class *Situation*):

$$\bar{S} = \lceil \dot{s} | \dot{s} \models Q \rceil \tag{3}$$

D. Situation Derivation and Relevant Information Reasoning

Situation theory also provides the notion of *abstract situation*. It is a collection of infons that are supported by a given situation, s:

$$S_s = \{\iota | s \models \iota\} \tag{4}$$

As explained in [15], STO, and thus STO-L, approximate Situation Theory by capturing the *supports* ($\models$) relation with an *entails* (or *derives*) relation (in the logical notation the symbol $\vdash$ is used for this). Therefore, the $supports$ relation between a situation and a collection of infons should be inferable from the knowledge of the agent, i.e., from the abstract situation.

Assume the agent was able to derive using its knowledge, A, that s holds. Denote the cylinder S_Q in Figure 4 as the knowledge in agent's knowledge base that is necessary to prove (or derive) that query Q (situation s) holds. This would be the *relevant information* to this situation.

E. Derivation of Relevant Information

Now we address the question of how to infer which facts belong to S_Q. Notice that if S_Q is sufficient to derive Q and since S_Q is a subset of A, then A should entail Q, too. The identification of the exact boundary of the relevant information S_Q is the key issue here. We need three steps to pick relevant information out of A.

Firstly, we only consider such subsets of A that are abstract situations denoted as S in Figure 4, which is a formal representation of situation s in the real world. However, not all abstract situations are necessary for answering the query. We only take a minimal subset of A that entails Q, or simply r_Q. These conditions can be captured by:

$$S_Q = \bigcap_{|S| \subseteq A} S \vdash r_Q \tag{5}$$

In STO-L, we create subclasses for *Situation* class to recognize different situation types. Moreover, STO-L imposes some restrictions on the instances of these classes. For instance, different types of situations support different infons, and for every situation there must be at least one relevant individual and at least one relevant relation. All such constraints limit the collection of facts that are considered in the intersection of Eq. 5.

We should also notice that the definition of minimal subset of relevant information in Eq. 5 does not guarantee the uniqueness. It means that there may be different collections of relevant information that have the same amount and entail the same query, but have different content.

V. A Cyber Attack Scenario Example

In this section, we consider a Common Gateway Interface (CGI) buffer overflow attack scenario which comes from the Skaion Corporations Advanced Research and Development Agency (ARDA) Testbed Dataset [26].

A. Scenario Description

CGI is a commonly used application framework, which is implemented on a Web server and provides an interface between the Web server and programs that generate the Web content. It is used for allowing limited access to the Web content information. In this scenario, a buffer overflow exploits against a CGI script of a petition website. People's "signatures" are all stored in a MySQL database. After the script establishes its connection to the database, it is exploited by a buffer overflow, which allows the attacker to query all information from that database, including the content that they should not be able to access. This attack has two steps:

- Attacker passes an overflow string to a CGI script on www.bprd.osis.gov.
- CGI is exploited, querying all tables in the MySQL DB it connects to.

B. Skaion Dataset

Skaion designed its testbed after the Open Source Information System (OSIS) [27]. The Skaion dataset about CGI overflow scenario consists of about 25 minutes capture from three different network sensors. These files include network traffic entering and leaving the entire internal network captured by tcpdump, the signature-based network intrusion detection system (IDS) alerts captured by Snort and Dragon, and FTP/Web sever logs. Besides, the dataset also provides knowledge of "ground truth", in which the actual roles of each IP address are listed.

This scenario only includes single stage attacks, which means it only has a simple scan or exploit or data exfiltration in this scenario. Besides the main attack, there are other background attacks (none of which are successful) and scans.

VI. Information Relevance Reasoning Experiments

In this section, we describe on the use of an implementation of our situation assessment framework for cyber intelligence information relevance reasoning on the Skaion dataset about an CGI overflow scenario.

A. Relevance Reasoning Work Flow

Figure 5 shows the workflow of cyber intelligence information relevance reasoning in our situation assessment framework. As illustrated in Figure 3, the cyber security analysts' queries and raw data from network sensors are the two inputs to the situation assessment framework. In our experiments, we manually map the query examples expressed by natural language to the *Infon* class of STO-L. The network raw data in Skaion dataset includes packet traffic, IDS alerts and operating system logs. We only import the Snort alerts to the STIX ontology as instances using Java to form the agent's knowledge base. We use Protégé [28] as the ontology editor.

The rest of the steps in this workflow are implemented by ontology based inference. In particular, we use BaseVISor as the inference engine. It allows for including additional inference rules.

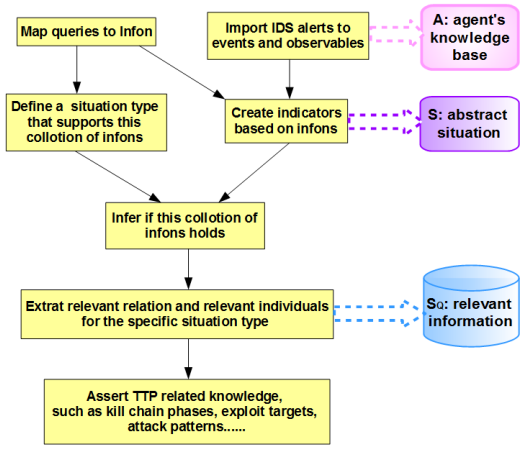

Fig. 5. Relevance Reasoning Workflow

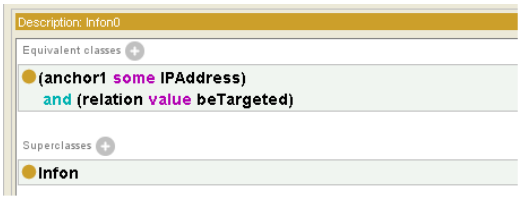

Fig. 6. Infon of Query 1

B. Query 1: Which servers have been targeted?

1) Infons: We interpret this query as the following infon:

$$<< beTargeted, IPAddress >>$$

where *beTargeted* is an unary relation, and *IPAddress* represents a collection of IP addresses of the servers being targeted. In our cyber situation assessment ontology, we represent this query as a subclass of *Infon* which was defined by a restriction as shown in Figure 6. All the instances of *Infon0* class should satisfy this restriction.

2) Situation type: We define a situation type named *TargetedServerSituation* that supports the infon above:

$$TargetedServerSituation \models << beTargeted, IPAddress >>$$

which is represented as a subclass of Situation by the restriction in Figure 7. All the instances of *TargetedServerSituation* class should satisfy this restriction.

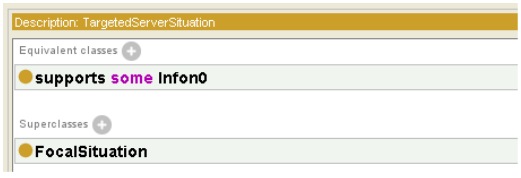

Fig. 7. Situation Type of Query 1

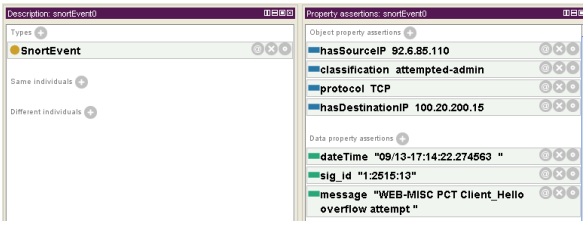

Fig. 8. An Example of Snort Alert Event Instance

It should be noted that both infon and situation type are defined manually with expert knowledge of cyber security domain.

3) Indicator rule: The core of Snort alert is a simple plain text message with a brief description of the event. Firstly, we parse the alerts and import the semantics to the ontology as instances using Java. Figure 8 is an example of Snort alert event instance.

We can see that an event instance carries several pieces of information associated with both object properties and datatype properties. The indicator rule is used to derive patterns or behaviors that indicate some aspects of a cyber threat. Since the infon gives us the focus of the situation assessment, we create the indicators based on the infon. For this query, the indicator rule is: create indicator instances for the events whose source IP address is from external network and the destination IP address is from internal network.

4) Be targeted rule: This rule is used to infer if the infon holds. If there are alert events in which source IP address is from external network and the destination IP address is from the internal network, then we assert that these destination IP addresses have been targeted.

5) Supports rule: If there are IP addresses that are targeted, we create instances of *Infon0* class (e.g., i_1) and corresponding instances of the *TargetedServerSituation* class (e.g., s_1), and assert that s_1 *supports* i_1.

6) Relevant information rule: The relations and anchors of the instances of *Infon0* will be asserted as relevant relations and relevant individuals of the targeted server situation.

C. Query 2: Are there CGI buffer overflow attack attempts?

1) Infons: We interpret this query as the following compound infon:

$$<< contains, CGIOverflowKillChain, KillChainPhase >> \\ \wedge << killChainPhase, Indicator, KillChainPhase >>$$

where *constains* and *killChainPhase* are two binary relations. This compound infon means the *CGIOverflowKillChain* contains some kill chain phases which are derived from some related indicators. Figure 9 shows the definitions of two elementary infons.

2) Situation type: We define a situation type named *CGIOverFlowSituation* that supports the compound infon:

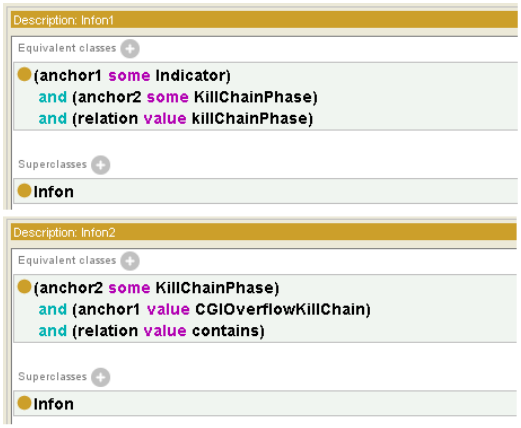

Fig. 9. Infons of Query 2

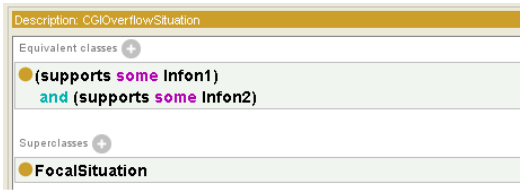

Fig. 10. Situation Type of Query 2

$$CGIOverFlowSituation \models \\ << contains, CGIOverFlowKillChain, KillChainPhase >> \\ \wedge << killChainPhase, Indicator, KillChainPhase >>$$

Figure 10 shows the definitions of this situation type.

3) Indicator rule: The signature of some Snort alerts has a CVE [29] reference associated with it, which is used as a key to get additional information about the particular vulnerability. However, in this dataset, most alerts don't have CVE references. So we need to analyze the signature of alert to recognize the attack type. For this query, we create indicator instances for the events whose signatures are related to the CGI buffer overflow kill chain phases.

4) Kill chain phase rule: Based on the scenario description, we can roughly define that the kill chain of this scenario contains two phases: overflow CGI phase, and query protect database phase. If the indicators have the same source and destination IP address, we assert that these indicators belong to the same kill chain phase.

5) Contains rule: If there are kill chain phases related to the buffer overflow indicators, we assert that this kill chain phase is contained by the *CGIOverFlowKillChain*.

6) Supports rule and relevant information rule: These two rules are similar to the rules in Query 1.

D. Inference Results

The total size of the Snort alerts files is 37 KB, which include 238 events. We import theses events into our ontology, and create an observable for each event. The rules described

TABLE I
INFERENCE RESULTS

	Precision	Recall	Input Information		Output Information	
			number of *individuals*	number of *properties*	number of *individuals*	number of *properties*
Query 1	90.0%	100%	472	1920	41	101
Query 2	100%	82.9%	472	1920	106	344

above were implemented in the BaseVISor rule language. Table I shows the inference results.

As depicted in Figure 3, the outputs of our framework contain two parts: the answers for the queries, and the minimal and sufficient set of information for answering the queries. So we first calculate the precision and recall of BaseVISor rule based inference. Then, we compare the amount of input information and output information.

E. Limitations

The main limitation of our experiments is caused by the input network data. This paper is just a prototype of the information relevance mechanism. To simplify the workflow, we only import a small part of the Skaion dataset (only Snort alerts) into our ontology. As we all know, all IDSs produce false positives and negatives. That means we need to consider all the dataset together, especially the packets traffic data, which is the most basic and trustful data to validate the alert events. We will extend our mechanism for diversity inputs in the future work.

Another limitation exists due to the format of ground truth in the Skaion dataset. The data created by Skaion is "labeled" according to the roles played by individual IP addresses (attacker, background attacker, background scanner, victim, server, etc.), instead of according to if individual packets were specifically malicious. So we just assume that every packet sent by a malicious IP address is a malicious packet. In the real situation, this assumption is not always true. In the future work, we need to figure out a more sophisticated way to characterize the behavior of adversaries.

VII. CONCLUSION

In this paper, we described a situation assessment framework for cyber security information relevance reasoning. We leverage the STIX ontology to represent cyber threat information, and extend it using situation theory. The information relevance reasoning mechanism we proposed can recognize specific situation types and identify the minimal and sufficient information for answering a query automatically. We implemented our framework on Skiaon dataset. The results show that our mechanism performed well. However, our experiments have some limitations. We will improve our framework in the future work.

REFERENCES

[1] [Online]. Available: http://www.dhs.gov/topic/cybersecurity

[2] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 37, no. 1, pp. 32–64, 1995.

[3] M. R. Endsley *et al.*, "Theoretical underpinnings of situation awareness: A critical review," *Situation awareness analysis and measurement*, pp. 3–32, 2000.

[4] J. Salerno, M. Hinman, D. Boulware, and P. Bello, "Information fusion for situational awareness," DTIC Document, Tech. Rep., 2003.

[5] P. Barford, M. Dacier, T. G. Dietterich, M. Fredrikson, J. Giffin, S. Jajodia, S. Jha, J. Li, P. Liu, P. Ning *et al.*, "Cyber sa: Situational awareness for cyber defense," in *Cyber Situational Awareness*. Springer, 2010, pp. 3–13.

[6] E. W. Burger, M. D. Goodman, P. Kampanakis, and K. A. Zhu, "Taxonomy model for cyber threat intelligence information exchange technologies," in *Proceedings of the 2014 ACM Workshop on Information Sharing & Collaborative Security*. ACM, 2014, pp. 51–60.

[7] A. C. Squicciarini, G. Petracca, W. G. Horne, and A. Nath, "Situational awareness through reasoning on network incidents," in *Proceedings of the 4th ACM conference on Data and application security and privacy*. ACM, 2014, pp. 111–122.

[8] V. Dutt, Y.-S. Ahn, and C. Gonzalez, "Cyber situation awareness modeling detection of cyber attacks with instance-based learning theory," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 55, no. 3, pp. 605–618, 2013.

[9] J. Strassner, J. Betser, R. Ewart, and F. Belz, "A semantic architecture for enhanced cyber situational awareness," in *Secure and Resilient Cyber Architectures Conference*, 2010.

[10] "STIX." [Online]. Available: https://stix.mitre.org/

[11] "CybOX." [Online]. Available: http://cybox.mitre.org/

[12] "MAEC." [Online]. Available: http://maec.mitre.org/

[13] "CAPEC." [Online]. Available: http://capec.mitre.org/

[14] "TAXII." [Online]. Available: http://capec.mitre.org/

[15] M. M. Kokar, C. J. Matheus, and K. Baclawski, "Ontology-based situation awareness," *Information fusion*, vol. 10, no. 1, pp. 83–98, 2009.

[16] B. E. Ulicny, J. J. Moskal, M. M. Kokar, K. Abe, and J. K. Smith, "Inference and ontologies," in *Cyber Defense and Situational Awareness*. Springer, 2014, pp. 167–199.

[17] "Owl web ontology language overview." [Online]. Available: http://www.w3.org/TR/owl-features/

[18] "STIX Ontology." [Online]. Available: http://www.vistology.com/ont/STIX/STIX.owl

[19] "Geonames ontology." [Online]. Available: http://www.geonames.org/ontology/documentation.html

[20] C. J. Matheus, K. Baclawski, and M. M. Kokar, "Basevisor: A triples-based inference engine outfitted to process ruleml and r-entailment rules," in *Rules and Rule Markup Languages for the Semantic Web, Second International Conference on*. IEEE, 2006, pp. 67–74.

[21] J. Barwise, "Scenes and other situations," *The journal of Philosophy*, pp. 369–397, 1981.

[22] J. Barwise and J. Perry, "Situations and attitudes," *The Journal of Philosophy*, pp. 668–691, 1981.

[23] J. Barwise, *The situation in logic*. Center for the Study of Language (CSLI), 1989, vol. 4.

[24] K. Devlin, *Logic and information*. Cambridge University Press, 1995.

[25] ——, "Situation theory and situation semantics," *Handbook of the History of Logic*, vol. 7, pp. 601–664, 2006.

[26] "Skaion." [Online]. Available: http://www.skaion.com/

[27] "Open source information system." [Online]. Available: http://fas.org/irp/program/disseminate/osis.htm

[28] "Protégé." [Online]. Available: http://protege.stanford.edu/

[29] "CVE." [Online]. Available: http://cve.mitre.org/