Leveraging cyber threat intelligence for a dynamic risk framework

Automation by using a semantic reasoner and a new combination of standards (STIX™, SWRL and OWL)

R. Riesco · V. A. Villagrá

Abstract

One of the most important goals in an organization is to have risks under an acceptance level along the time. All organizations are exposed to real-time security threats that could have an impact on their risk exposure levels harming the entire organization, their customers and their reputation. New emerging techniques, tactics and procedures (TTP) which remain undetected, the complexity and decentralization of organization assets, the great number of vulnerabilities proportional to the number of new type of devices (IoT) or still the high number of false positives, are only some examples of real risks for any organization. Risk management frameworks are not integrated and automated with near real-time (NRT) risk-related cybersecurity threat intelligence (CTI) information. The contribution of this paper is an integrated architecture based on the Web Ontology Language (OWL), a semantic reasoner and the use of Semantic Web Rule Language (SWRL) to approach a Dynamic Risk Assessment and Management (DRA/DRM) framework at all levels (operational, tactic and strategic). To enable such a dynamic, NRT and more realistic risk assessment and management processes, we created a new semantic version of STIX™v2.0 for cyber threat intelligence as it is becoming a de facto standard for structured threat information exchange. We selected an international leading organization in cybersecurity to demonstrate new dynamic ways to support decision making at all levels while being under attack. Semantic reasoners could be our ideal partners to fight against threats having risks under control along the time, for that, they need to understand the data. Our proposal uses an unprecedented mix of standards to cover all levels of a DRM and ensure easier adoption by users.

Keywords STIX™ · SWRL · OWL · Cybersecurity · Dynamic risk management (DRM) · Cyber threat intelligence (CTI)

1 Introduction

Motivation

Current frameworks and methodologies for risk assessment (RA) processes [1,2] follow an iterative approach in which a partial snapshot of the organization assets and business processes is periodically taken for the estimation of its risk exposure and it is primarily based on expert and subjective theories. Risk management (RM) and its countermeasures, are also identified during these periodical reviews with the intent to keep the risk below an acceptable level.

On the other hand, cyberthreat landscape and the attack surface of any organization change constantly. Then, its corporate risk level (which is based on probability and potential impact of such threats) changes too. This real and dynamic behavior renders these legacy frameworks and methodologies highly ineffective and unreliable for any organization or risk analyst.

None of probability and impact variables used for risk estimation can depend just on a specific person experience. Experience is relevant and necessary but it is not enough, and we should take into account the real threats in order to address them by investing in countermeasures as earlier as possible at the same time we measure their effectiveness. There is just one reality of an organization but current approaches, even by international recognized organizations in the cybersecurity arena, are not addressing RA/RM processes by considering NRT (near real time) threats.

It can be worse because of the adoption of the Internet of Everything (IoE) as it multiplies the number of devices, data, connections and processes as never expected. The more number of assets to protect, the larger attack surface, so increasing automation is encouraged. It does imply a more complex and heterogeneous context. Discovery protocols and processes play an important role; however, they lack of standardization.

Unintentional incidents might increase as well, while more complex contexts are considered. New risk measures have to be included during change management processes in order to prevent either unintentional or intentional incidents before they happen. At the same time, a small cybersecurity flaw is usually enough for an attacker to success. Furthermore, it can be easier for an attacker when such a flaw is not part of the RA/RM scope, as it could lack of specific countermeasures.

As an example, social engineering attacks are based on people cheating other people in order to overcome specific countermeasures of the organization. An attacker can directly get access to a specific password during a phone call with the victim, and this is usually easier than launching a specific cyberattack. The victim will simply provide it to the attacker if it supposes to be talking to the IT admin or any other trusted person. Other example is to bypass the perimeter by providing a malicious USB stick outside of the perimeter to a victim, who will potentially connect it inside the perimeter despite all efforts and investments made by the organization at the perimeter level.

In our case, a top manager (with enough access rights to classified data) is usually reading a specific third-party news Web site on a daily basis. Web surfing to these external systems is not part of the organization RA/RM scope. Then, despite all efforts at the perimeter and risk management countermeasures, a watering hole attack could affect not only the victim data and its laptop but also any data accessible by the victim like the classified data. Furthermore, usually risk assessments (as demonstrated in our work by the example of an international lead organization), are not updated dynamically once a security event, an attack or an incident is detected. RA/RM processes are usually disconnected from incident handling or from any other real-time cybersecurity threat intelligence system. It also implies limitations to have more effective prevention safeguards, better detection mechanisms and more real ROI (return of investment) calculations over safeguards.

On the other hand, cyber threat intelligence domain has its own challenges as well. Until today there is an asymmetric battle, common vendors try to fight against the unknown by using products that were designed only to fight the known threats, that is, using IoCs (indicators of compromise) of threats. At the same time, there is a need of a common language as most vendors use vendor specific taxonomies along their data models. Sometimes, the reason provided is related to efficiency; however, all vendors realized the importance of information sharing as they cannot fight alone against increasing unknowns. They realized that having additional and reliable information, coming from any other partner or external source, will benefit their own customers. Their own intelligence data will be then enriched by additional and external data. In order to share data efficiently, there is a need of a common language, if possible, based on standards like STIX™.

IoCs (indicators of compromise) like hashes, IP addresses, domains and vulnerabilities are in the bottom part of the "Pyramid of Pain" [3] which means that, if filtered or blocked, the threat actor will easily overcome those filters by simple tasks. On the other hand, we would be in the top of the pyramid when our defense is based on filtering the TTP of the attack. That will cause enough "pain" to the attacker, that is, if our measures are very difficult to overcome by the attacker, it will certainly decide to give up the attack, or at least it will have to completely change the whole TTP which is a complex and difficult task and it can have high associated costs. If vendors are responding to attacks by blocking certain hashes, IP or domains, attackers will make lateral and easy movements to change such known artifacts into new (unknown) ones. It is quite easy for threat actors to modify such variables. If we really want threat actors to give up, we need to force them to change their TTP (techniques, tactics and procedures). Then, **solutions should be focused to filter sequences, patterns and behaviors instead of static IoCs (indicators of compromise)**.

Threat intelligence specification drafts like CybOX™, STIX™or TAXII™by OASIS, between others, are becoming de facto standards with great involvement from the community willing to share threat data. Current implementations of such protocols evolved from XML schema to JSON format in v2.0 [4]; however, their authors are proposing a potential future research direction into a more expressive standard (like semantic RDF/OWL mentioned at implementations' section of [5]) as there are still important limitations to describe more complex concepts like TTP (tactics, techniques and procedures) [6], campaigns [7] or incidents [8], between others.

Information sharing nowadays is more than just a proactive individual initiative; as an example, the Directive on Security of Network and Information Systems (the NIS Directive) [9] was adopted by the European Parliament on 2016, July the 6th. In its Communication of 2016, July the 5th, the European Commission encourages member states to enhance cross-border cooperation in case of a major cyber-incident. The directive establishes a baseline for a formal cooperation between member states and beyond; however, confidentiality, data protection and national security must be guaranteed, while incident information is being shared between interested parties. National Competent Authorities and/or CSIRTs (Computer Security Incident Response Teams) are empowered by the directive to assess essential services operators and digital service providers risk level exposure. They are also forced to notify relevant incidents as a mandatory legal requirement. The directive is also open for voluntary information sharing to other types of organizations in case of significant impacts caused by cyberattacks.

With regard to standardization, article 19 of the directive promotes convergent implementation without imposing or discriminating in favor of the use of a particular type of technology. It encourages the use of European or internationally accepted standards and specifications. Now, being information sharing mandatory, the same cybersecurity concepts at European level (related to risk and threat intelligence domains) are probably in risk of not having the same understanding around the globe.

In this work, we propose a mix of 3 standards to overcome all described limitations: STIX™ [5] as an industry-driven standard, as well as OWL [10] and SWRL [11] to overcome all semantic expressiveness and limitations of STIX [6–8]. We consider that if our proposal is based on standards, it will be easier to implement and deploy by several organizations in the future.

Semantic ontology [12], supported by scientific language researchers as well as the World Wide Web Consortium (W3C), is a standard solution to create formal models, defining concepts, domains and relationships (even complex ones) guaranteeing unequivocal meaning.

They are considered building blocks for semantic inference [13], which is a mechanism to discover new relationships, to automatically analyze the content of the data and to manage knowledge. These inference-based techniques are also important in discovering possible inconsistencies in the (integrated) data. The role of ontologies is to help data integration when, for example, ambiguities may exist on the terms used in the different datasets, or when a bit of extra knowledge may lead to the discovery of new relationships. Consider, for example, the application of ontologies in the field of health care. Medical professionals use them to represent knowledge about symptoms, diseases and treatments. Pharmaceutical companies use them to represent information about drugs, dosages and allergies. Combining this knowledge from the medical and pharmaceutical communities with patient data enables a whole range of intelligent applications such as decision support tools that search for possible treatments, systems that monitor drug efficacy and possible side effects, and tools that support epidemiological research.

We selected the OWL Web Ontology Language [10] which is designed for the use of applications that need to process the content of information instead of presenting information to humans. OWL facilitates a greater machine capacity of interpretation of Web content than the one supported by XML, RDF and RDF schema (RDF-S). It is because it provides additional vocabulary along with formal semantics. OWL has three increasingly expressive sublanguages: OWL Lite, OWL DL and OWL Full.

Semantic Web needed a separate language due to the nature of its applications. Interoperability is one of the primary goals of the Semantic Web, and there is a significant interest in its standardization.

The goal of sharing rule bases and processing them with different rule engines has resulted in RuleML, SWRL, Metalog, and ISO Prolog, and other standardization efforts. One of the key steps to rule interoperability on the Web is SWRL [11] which was designed to be the rule language of the Semantic Web. SWRL is based on a combination of the OWL DL and OWL Lite sublanguages of the OWL Ontology Web Language, the Unary/Binary Datalog (Datalog is a query and rule language for deductive databases that syntactically is a subset of Prolog) and Sublanguages of the Rule Markup Language.

SWRL permits users to write hornlike rules expressed in terms of OWL concepts in order to reason about OWL individuals. The rules can be used to get new knowledge from already existing OWL knowledge bases. The SWRL specification does not impose restrictions on how reasoning should be performed with SWRL rules. Thus, developers are free to use a variety of rule engines to reason with the SWRL rules stored in an OWL knowledge base.

In **SWRL** [11] each rule has an antecedent (body) and a consequent (head). Once all conditions in the antecedent are verified, all the consequent conditions are also fulfilled.

"antecedent -> consequent"

As an example, in the following SWRL rule:

```
hasParent(?x1,?x2) AND hasBrother(?x2,?x3) ->
-> hasUncle(?x1,?x3)
```

If two individuals ?x1 and ?x2 have a relationship where ?x1 has a parent ?x2 and, at the same time, ?x2 has a brother ?x3, then ?x1 will have an uncle which is ?x3.

From this rule, if John has Mary as a parent and Mary has Bill as a brother, then John has Bill as an uncle.

Variables used in consequent (in our case: ?x1, ?x3) have to be defined in antecedent.

Atoms in SWRL can be of the form C(x), P(x,y), sameAs(x,y) or differentFrom(x,y), where C is an OWL description, P is an OWL property, and x,y are either variables, OWL individuals or OWL data values.

In SWRL [11], there are different types of atoms to express different meanings:

- belonging to an instance (a variable can be used instead) to a class extension,
- a literal to a data type listed in OWL DL,
- a relation between two instances of object type through a property of type ObjectProperty,
- relationship between a copy of type object (in the subject position) and a literal (in the object position) through a property of type DatatypeProperty,
- or equality and inequality between two copies.

SWRL [11] *increases the OWL expression ability to define rules and restrictions* It includes a high-level abstract syntax for conditional rules (Horn-like rules) in both the OWL DL and OWL Lite sub languages of OWL. SWRL allows defining complex conditions to be fulfilled in the antecedent of the rules, through the built-ins, the AND operator and the use of atoms. The use of variables in atoms allows defining constraints that are not possible in RDF or in OWL.

Taking into account that we developed the complete ontology version of STIX™v2.0 as well as a complete DRM ontology, there are several options to create enriched antecedent (body) and enriched consequent (head) rules as seen in this work.

Reasoners [14] are tools than can perform automatic and continuous reasoning tasks like inferencing, deriving new facts from existing ontologies and guaranteeing data consistency. They are based on RDF, OWL [10] or some rule engine like SWRL [11].

We selected Pellet incremental reasoner [15] for our work, based on Ontology Web Language (OWL) and SWRL to manage all our cybersecurity threat intelligence and risk data.

Interoperability, standardization, expressiveness as well as the need of automation are some of our reasons to propose a model based on ontologies to either minimize the impact of different threat and risk management interpretations among different countries; at the same time, it enables effective automation via machine to machine communication.

Related work Since 2007, different authors have been investigating the usage of ontologies for risk domain.

It is the case of Herzog et al. [16] where authors define a generic security domain ontology specified in OWL which covers most of the aspects of an information security domain. It provides a detailed vocabulary as well as it supports reasoning capabilities. It is built on classical risk assessment concepts: asset class, threat class, vulnerability class, countermeasure class, security goal class, defense strategy class. Authors provide with some detailed subclasses and relationships between them.

Fenz et al. in [17] contribute with ontologies for a quantitative risk analysis in which authors visualize the damage caused by specific threats, outage costs and the recovery time. Running the program with added safeguards shows their benefits and offers objective data for decision making: which safeguards to implement and to avoid installing countermeasures that are not cost-effective. Authors thus justify the need to have a security ontology to clarify the meaning and interdependence of unambiguous IT security relevant terms which then can be used to facilitate qualitative risk analysis and decision processes.

Fenz in [18] contributes with ontologies to define IT security metrics. Author plans to align it with ISO 27004 standards and to apply it in real-world audit scenarios as well as to go further in the degree of automation.

Fenz et. al. in [19] integrate an ontological information security concept in risk-aware business process management. The ontology is based on NIST, and authors provide threat, vulnerability and control sub-ontologies. Authors propose to improve and extend the threat classification in order to consider the threat for human life as the main priority in case of any risk. In addition to this, supplemental information to be considered by the ontology can provide valuable and essential details for decision makers.

More recent work like Villagra et al. [20] is used to propose a model based on ontologies to integrate and share alerts between different Security Information Management Systems.

Villagra et al. [21] propose an Automated Intrusion Response System (AIRS) based on ontologies, that is, to use reasoning to select optimum responses. The system will infer the optimum responses at network level (e.g., intrusion prevention systems). This time, they work at network domain.

Obrst et al. [22] introduce a proposed ontology for cybersecurity, especially as an extension of MAEC (Malware Attribute and Enumeration Characterization). Authors use as a reference the Diamond Model of Malicious Activity.

Singapogu et al. [23] describe a proposed ontology for making enterprise risk assessment by supporting the IT security risk analysis process.

Erbacher [24] developed a packet-centric ontology named PACO which allowed them to represent and capture the atomic elements of network communication, i.e., packets and sequences of packets. It is a proposed model as a basic for more holistic approaches.

Syed et al. [25] worked on an integration between STIX™and ontologies for situational awareness which is a very interesting approach. They demonstrated the benefits for different use cases (vulnerabilities associated with PDF readers, suggestion of similar SW, etc.) as a very interesting contribution, for example, to check the impact of changing vendors.

With regard to querying the data, some approaches [26,27] suggest semantic, but they are still semantic-agnostic nor using standards.

Meszaros et al. in [28] propose a framework for online service cybersecurity risk management applied to a large enterprise. The risk model is providing simplicity to manage by either providers or consumer's viewpoints. It is also aligned with standards [1,2].

One of the most recent as well as interesting works is the work of Qamar et al. [29]. Authors implemented the ontology version of STIX™(version 1.X), together with CVE and network ontologies to build STIX analyzer, a framework to perform data-driven analytics for threat intelligence and information sharing. It is based on known shared and threat data from threat repositories. One of its main use cases is working on attribution. Authors also provide a way to simulate and calculate some risks based on exposure levels. It is not focused on leveraging an organization threat intelligence data having a near-real-time detection, protection and risk management.

We propose a complete dynamic risk management framework compatible with any widespread risk assessment and management standard. We also provide the needed expressivity and granularity for risk management frameworks, for example, considering the differences between the likelihood of threats based on knowledge, access rights and behaviors of each of our users. Our proposal is also focused on behavioral detection of new and still unknown threats (when IOCs are still not available in any intelligence repository or feed) and complex TTP behaviors. We also work with the new version of STIX™2.0 which has several improvements from version 1.X, like an integration between CyBox™and STIX™.

Poolsappasit et al. in [30] proposed a very interesting dynamic risk management model based on Bayesian attack graphs, using conditional probabilities to encode the contribution of different security conditions during system compromise. They estimate an organization security risk from different vulnerability exploitations based on the metrics defined in the Common Vulnerability Scoring System (CVSS) [31].

Mozzaquatro et al. in [32] provide an interesting approach based on some of our building blocks like OWL, SWRL and a reasoner for detecting, identifying and classifying vulnerabilities (or bad configurations) of IoT devices. They propose 3 layers (design, run time and an integration layer). The reasoner is used to propose specific measures to improve vulnerabilities or bad configurations (e.g., if a WEP config is detected, then it is best to use WPA2). At the same time, authors use signature matching sensors and IMDEF format. However, it is not oriented to a risk assessment or management framework like our work, and it can potentially be connected with our framework to benefit from it. At the same time, our proposal goes beyond IoT, vulnerabilities and IMDEF. We implemented the whole STIX™ spec draft creating its ontology version to cover as much types of threats as possible leveraging from that standard draft. We also use STIX™ along the whole system (sensors work, understand and deliver STIX™ format data). We also propose a fully integrated domain of STIX™ threat and the risk domain. In our case, using SWRL and the reasoner we are able to use it for detection at sensors level, but we are also able to detect malicious patterns without knowledge of specific signatures. We create rules based on SWRL following a certain pattern to make reasoning proposals within risk management and CTI domains once those patterns have been detected. As a future research direction, authors in [32] propose the usage of artificial intelligence and Bayesian networks to overcome the limited detection capabilities of signature-based matching sensors. As demonstrated in our work, we consider that ontologies, SWRL and reasoners can be used together with STIX™ to handle the needed expressiveness to detect patterns of unknown IOCs.

Despite all references, no one is providing a solution to have a near-real-time dynamic risk framework based on dynamic threat detection. Even all of them are based on threat data (mostly IOCs), but they are not based on patterns or behaviors. Semantics has been used in some of the references provided even in very recent works as a good solution for the current lack of expressiveness.

Both domains need to be connected (cybersecurity threat intelligence and risk domains) completely, and the connection should be based on standards to enable risk and threat information sharing.

By using and developing the entire STIX™ new version 2 in OWL format, we are promoting the widespread industry-driven taxonomy for threats but in its semantic version to overcome their limitations with regard to expressiveness [5–8].

Inference [13] by our Pellet reasoner [15] will enable automatic threat and risk data discovery. At the same time, the usage of SWRL [11] rules will also enable the usage of extended and enriched algorithms.

As an example, we could provide the automation needed to detect unknown threats based on patterns like detecting any technique, tactic and procedure (TTP) [6] with enough expressivity. This kind of detection techniques will go beyond current approaches based on known IoCs (indicators of compromise). At the same time a detection is done, the system will infer risk re-calculations dynamically. SWRL rules will be able to give solution to any type of algorithm, and it can be a cybersecurity threat intelligence algorithm, a risk domain algorithm or a mixed CTI-risk one.

By using SWRL [11], we will also simplify the creation of either threat or risk algorithms specially for non technical people. A one-day training on domain ontologies and the usage of SWRL will be enough. Until today, those who create algorithms needed development skills [33].

Approach and results

Our approach is based on OWL ontologies [10] and Semantic Web Rule Language (SWRL) [11] as seen, for example, in Figs. 1 and 3, respectively. It provides a coherent

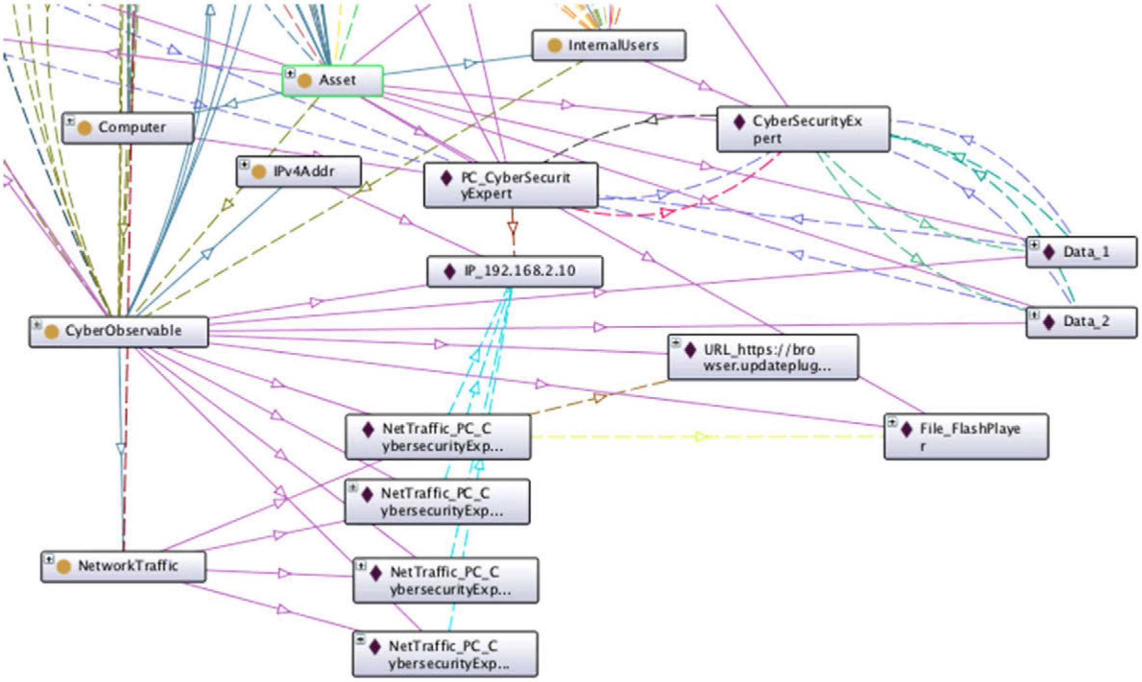

InternalUsers
Asset
CyberSecurityExpert
Computer
IPv4Addr
PC_CyberSecurityExpert
Data_1
IP_192.168.2.10
CyberObservable
Data_2
URL_https://browser.updateplug...
NetTraffic_PC_CybersecurityExp...
File_FlashPlayer
NetTraffic_PC_CybersecurityExp...
NetworkTraffic
NetTraffic_PC_CybersecurityExp...
NetTraffic_PC_CybersecurityExp...

Fig. 1 Ontology representation of our integrated (OWL and STIX™based) cyber threat intelligence and dynamic risk management architecture, e.g., CybersecurityExpert User instance, its data access, personal computer, CyberObservables, user-related NetworkTraffic related to a malicious dropper

and integrated solution to the needed expressiveness of such concepts and rules but also to inference [13] new knowledge. They could also resolve the lack of interoperability between existing risk management frameworks and methodologies under a common language and a common understanding in order to expand risk context into a global and more realistic picture.

Ontologies [12] are an explicit and formal way to represent concepts, their meaning and their relationships. Explicit because it defines concepts as classes, properties, relationships, functions, taxonomies, axioms as well as rules and restrictions. It is formal because it is defined by a language, which is interpretable by machines. It is a conceptualization, because it is an abstract model for a simplified view of the domain to be represented (e.g., structure). It is also shared by the community for consensus.

By connecting both domains (risk and cyber threat intelligence) semantically, we demonstrate the ability to dynamically assess organizational risks based on near-real-time (NRT) threat data.

We propose a model based on layers (data, business logic, services/applications and visualization) as well as to integrate both domains and ontologies to approach both domains simultaneously.

In order to test our proposal, we selected a specific international recognized organization in the cybersecurity arena. Its current static risk assessment and management (RA/RM) approach has been improved by our model into a dynamic risk framework which is successfully reacting and responding to real-time threats.

We have simulated a well-known watering hole attack from July to August 2017 targeting our selected organization (however, it was not the target of the real attack) in order to test how our model provides with a more effective way to prevent, detect and respond against a watering hole TTP (techniques, tactics and procedures). Once the TTP is detected, security events are created. Those security events would trigger specific risk re-assessments dynamically.

Now that we are able to model any TTP with enough expressiveness, we would be able to model any TTP like those of [34].

Contributions (i) Layered architecture for dynamic risk assessment and management (DRA/DRM) based on STIX™, OWL ontologies, SWRL and a Pellet semantic reasoner. (ii) Evolution and integration of cyber threat intelligence data within DRA/DRM processes. For that, we implemented the OWL of a complete version of STIX™v2.0 [4] de facto standard (STIX2.owl ontology that was imported by our DRM.owl ontology). (iii) Definition of several SWRL rules as algorithms and axioms to support all the business logic made by the Pellet incremental semantic reasoner used in this work.

Paper organization In Sect. 1, we introduce the problem, the motivation, related work as well as the summary of our approach and main contributions.

In Sect. 2, we give details about the problem and how the selected (and internationally recognized) organization is addressing each domain by giving partial details of the methodology used as well as each of the layers involved (data, business logic, services/application, visualization). We also make a detailed description of our proposal by taking into account all specificities of the selected organization inside.

In Sect. 3, we describe the selected attack (TTP and motivation) to test how our proposal would benefit the selected organization threat detection, risk assessment and risk management processes dynamically and in an integrated way. In this case, we test how it performs under an attack situation of type watering hole, which are usually very difficult to detect. Our framework will implement pattern detection techniques without their indicators of compromise (IoCs). At the same time, the integrated approach will enable an automatic reassessment of our organization cybersecurity risks triggered by each particular (new) knowledge.

In Sect. 4, we describe the implementation.

In Sect. 5, we describe our main results.

In Sect. 6, we describe our main conclusions.

2 The problem and our proposal

We present the cyber threat intelligence (CTI) model considered in this work also as an evolution of current offering in the market. Main purpose of threat intelligence is to have as much sophisticated knowledge as possible about unknown threats either for having better detection capabilities or having better prevention ones. For example, an emerging threat which is affecting similar entities would be able to affect our organization with high probability if we share the same threat actors' motivation. Detecting this type of complex connections by inference and expressiveness will allow us to provide enhanced prevention and detection services.

Automation is needed due to the evolution in the number of threats, services to protect, lack of resources and complexity of each organization. Some of the most relevant and common cybersecurity needs with regard to automation are related to:

- Enrichment of threat information context.
- Security event detection.
- Security incident detection and prevention.
- Incident triage by severity (incident handling).
- Information sharing control (why, what, when and how).
- Risk assessment and risk management.

In addition to this, incidents and risks are usually managed by different teams due to the specificity of each domain and levels involved (operation tactical or strategic levels), and their tools are in most of the cases not integrated as they should. Threat intelligence data are sometimes connected to incident handling; however, none of them are connected to enterprise risk management by default.

Each domain has its specificity also with regard to the concepts they manage. There are different initiatives either in cyber threat intelligence (CTI) [22,25] or risk management (RM) [16–18,23] trying to define clear taxonomies of each domain, but all of them in an isolated way. They are only targeting one domain at the same time, and in most of the cases, it is a partial approach. Querying approaches [26,27] are suggesting semantics, but they are semantic-agnostic nor using standards.

The most relevant and advanced products in CTI are started to use open standards like STIX™ that will allow them to share threat intelligence information based on a common taxonomy that is accepted by the majority of the industry as a standard format. STIX™ is offering real benefits by its taxonomy of cybersecurity threat domain concepts but also defining relationships between their objects. However, some concepts inside STIX™ cannot be represented yet, due to their limitations to describe more complex concepts like TTP (tactics, techniques and procedures) [6], campaigns [7] or incidents [8]. STIX™ version 2 has evolved into a more integrated approach (e.g., cyberobservables are represented inside) as well as it uses JSON instead of XML (used in STIX™ v1.1) for better automation with current product offering. It still lacks full expressiveness; however, v1.1 white paper [5] suggested RDF/OWL as a potential future solution for it.

Despite this evolution, we need formal models to describe meaning of even complex concepts that cannot be described by today. At the same time, formal models would enable machines to also understand those concepts and relationships for a better and more effective automation [12,13].

In this work, we introduce the evolution of STIX™v2.0 JSON [4] into a semantic STIX™2.0 OWL (ontology) version and associated SWRL (Semantic Web Rule Language) rules as a formal model of a CTI framework to solve above needs.

On the other hand, risk assessment (RA) and risk management (RM) domain is using a different taxonomy to describe the type of risks, asset dependencies but also the appropriate controls (safeguards or countermeasures). We implemented and used the international standard ISO2700X family [1] as well as the ISO31000 [2]. Because there are different reference standard models, the creation of a formal model would also enable us to have interoperability between themselves.

Another relevant factor that could be one of the most important reasons for not having any effective integration between RA/RM and CTI frameworks yet is that both domains are using different timing.

RA/RM is usually using Deming cycle (plan, do , check, act) as an improvement life cycle that is not real time, which is different from the reality of cyber threat intelligence and incident handling as they have to be at least NRT (near real time) or if possible, RT (real time).

Not having NRT/RT risk assessment and risk management nowadays mean that our management is not well informed of their exposure levels on time as well as investments are based usually on annual external consultants estimations (e.g., standard threat probability estimations) not considering the reality of our organization's threat landscape. Risk calculations are then only taking place few times per year. Mostly, one per cycle.

Dynamic risk assessment (DRA) and dynamic risk management (DRM) are new concepts to describe the real need to have knowledge of our risk exposure level close to NRT/RT.

In this work, we also introduce a new semantic DRA/DRM OWL (ontology) and associated SWRL rules leveraging CTI as a formal model of a DRA/DRM framework to solve all problems described above.

The main contribution of the present work would be the integrated (CTI + DRA/DRM) architecture to cover all above needs of different domains at the same time. It is based on ontologies, SWRL rules and the use of a reasoner. All domains are now integrated with regard to concepts (ontologies STIX2.owl and DRM.owl) but also with regard to timing (NRT/RT).

2.1 Organization under study: a National CSIRT

Here we introduce the organization selected to test our work. We selected a National CSIRT (Computer Security Incident Response Team) as we consider it an international reference for its corporate maturity (risk management framework certified under ISO and under a National Cybersecurity Scheme) and the maturity of their CSIRT services (incident prevention and incident response services based on cyber threat intelligence services).

This public organization is responsible to provide nationwide preventive and incident response services affecting citizens and private sector audience, including critical infrastructures.

At the same time, this organization has its own SOC (security operation center) to protect its corporate IT infrastructure. As described above, it is certified under ISO2700X Information Security Management standards but also under a new National Risk Management Standard which is a must for public entities like our selected organization.

Several attacks are targeting our organization (e.g., hacktivism, state-sponsored, etc.) dynamically. Security events are received and handled by its SOC; however, there is still not connection between those threats and its risk management framework.

2.1.1 Cyber threat intelligence model of the organization under study

Here we describe the current CTI model of the selected organization based on layers (bottom-up).

With regard to the data layer, the organization has different international agreements with partners as well as different providers to gather valuable threat information affecting their customers. Several IOCs (indicators of compromise) are received from external sources with regard to different threats. Some examples are: malicious URL, domain, subdomains, IP addresses, bots, botnet servers, defacements, vulnerable assets, vulnerabilities, malware samples, hashes, SPAM emails, phishing campaigns.

The organization has its own threat data coming from their advanced sensors: honeypots (low, medium but also high interaction ones), dark/deep Web monitoring systems, and so forth.

Threat data information (internal and external) is parsed into its own data model through a SIEM (Security Information and Event Management), and it is persisted on a Big Data architecture (Apache Spark + Hadoop). SIEM vendor provides a restricted proprietary data model so there is a need to persist a parsed version into the Big Data under STIX™-format.

Each organization will have a different CTI dataset depending on its own interests CTI data will range from their own collection of data to external interested data from business partners/feeds. As an example, we consider a CTI dataset for our work that includes different types of data like:

- Incident handling attacks received.
- Intelligence analyst data.
- Other internal data like strange patterns of traffic of specific users.
- External CTI data provided by third parties (e.g., supply chain, business partners, other CSIRT, equivalent entities within the sector, etc.) but in the same format (STIX™).
- Any data of any STIX™ object/concept.

The idea is to keep a coherent dataset in a semantic version of STIX™to be able to run SWRL rules. We are really open to any type of possibilities but limited to that taxonomy.

The business logic layer is formed by algorithms and rules. Today, logic is widespread between sensors, security event information management (SIEM), as well as in the Big Data. It has mainly three problems: isolation (data not shared, API not available, proprietary language), limitations (static, predefined and not touring complete logic) and complexity [33] (SCALA language for Big Data, vendor specific for SIEM).

Services/application layer is where actions (e.g., alerts, notifications, etc) are taken to cover all potential use cases (incident prevention, incident detection notification, incident handling, information sharing, KPI updates being triggered, etc). It uses the business logic and data layers. As an example, the SIEM is used to open incident tickets automatically into their incident response ticketing system for its NRT/RT capability.

For example, the information sharing application helps partners and stakeholders of our selected organization. TLP (traffic light protocol) tag is used to keep confidential information safe, and it is equivalent to an ACL (access control list). TLP avoids cybersecurity information to be shared without consent; at the same time, we benefit the whole cybersecurity community when the info can be shared with a broader audience. Different levels will apply depending on the confidentiality of the information.

In case of the visualization layer, the organization is currently migrating into a business intelligence platform to guarantee effective visualization to the three different levels (operation, tactical and strategic levels). Nowadays visualization is provided only to operational level via Web access to data storage (SIEM logger and Big Data). Tactical and strategic levels have been approached partially, still without business intelligence, flexibility to deploy new algorithms and interactive drill down options to explore the data. This is something that we will have by definition by using ontologies (e.g., graphs). As data model is not completely standard yet (still using some vendor specific data model), tactical and strategic levels could experience some problems when new indicators need to be provided. In this case, there are situations where data meaning is not clear enough (e.g., compromised resources versus incidents). By using ontologies [12], we will address these inconsistencies.

2.1.2 Risk assessment and risk management model of the organization

Here we describe our organization risk assessment and risk management model:

Risk assessment has limited scope as only a few (but business essential) services or projects are in the scope of the risk assessment process. Risk owners usually give the service or project context by filling down a form and/or an interview. The scope includes the context, that is to say, assets and dependencies either internal or external. The context also includes the compliance, legal, politics factors. Risk owners have to evaluate each service by different dimensions:

- Information security: confidentiality, integrity, accounting, authenticity, availability.
- Business impact assessment (recovery time objective, recovery point objective).
- Strategy.
- Satisfaction.

Dependencies are usually manually defined by risk owners which is considered a big limitation, as they are usually not experts on their corporate network topology to clearly define IT dependencies between assets (better known by IT staff). They evaluate the importance of each service with regard to different dimensions. That valuation is then considering all dependencies. As a result in a top-down approach, all assets inherit the valuation from above. An asset which is providing support to two different services in scope will have inheritance from both services. Services in risk assessment scope will depend on data, which at the same time will depend on SW/HW and later will depend on their users who will be using those assets.

Threat inventory is taking place again by hand, identifying most relevant threats that can threaten our service/project. Those threats can harm the image of our organization sometimes, so those risks have also to be manually entered. After that, safeguards have to be manually identified as well. They could partially mitigate our risks depending on the threat and the countermeasure.

Risk should be a function of probability and impact as suggested by most of the methodologies and standards today.

The organization under study decided to follow the following formula where risk is a combination of probability and impact, reduced by availability of certain countermeasures:

$$R_i = P_i + I_i - C1_i - C2_i \tag{1}$$

where

- R_i is the residual risk of threat i,
- $C1_i$ is the decreasing value of the impact (severity) I or Probability P of threat i due to a countermeasure 1 (specific control established in the organization),
- $C2_i$ is the decreasing value of the impact (severity) I or Probability P of threat i due to a new proposed countermeasure 2 (specific new control to be established),
- P_i is the probability P of threat i,
- I_i is the impact (severity) I of threat i when it is materialized.

Threats could be of different types (strategic, compliance, physical security, IT security, quality and process management, others).

Because the assignment of each safeguard to each threat is not really done in the organization today (it had to be done by hand and high level of granularity), the formula is simplified in the organization under study by using average values.

Depending on the new risk value after all new countermeasures are setup, risk is classified in the following scale (from 1 to 10):

- Extreme (8–10): risk is unacceptable,
- High (6–7): risk is undesirable,
- Medium (5): risk is tolerable,
- Low (1–2–3–4): risk is acceptable.

A mitigation strategy is recommended for extreme and high risks with different possibilities:

- to reduce,
- to avoid,
- to share,
- to transfer the risk.

An investigation strategy is recommended for medium risks with different possibilities: to assume or to reduce the risk.

A monitoring strategy is recommended for low risks where usually risks are assumed.

In addition to this, all strategies should take into account potential actions of different impact in different dimensions:

- Image and reputation.
- Compliance.
- Security.
- Budget and costs.
- Operations (Ops).

Other special response to risk strategies can also be considered where

- attack, deception, deterrent, information sharing, awareness, are some examples.

Summarizing, being the selected organization an international recognized CSIRT which is providing nationwide advanced cybersecurity services, it has its own risk as organization. Still CTI and RA/RM processes are isolated between themselves, the reason behind after some interviews is that they are supposed to belong to different domains and they are managed by different teams inside the organization (as usual in most organizations).

Data, business logic, services/applications and visualization from each of these domains are completely separated. RA/RM is using static tools (Excel, etc.) different from CTI tools, but they are also using different taxonomies, level access and a different timing, when they should not be isolated.

2.2 Our proposal: integrated CTI and DRA/DRM architecture

Here we introduce our proposal as an integrated and layered architecture.

2.2.1 Semantic data model

Here we created two main OWL ontologies (OWL—Web Ontology Language) for the two domains:

- STIX2.owl for all threat intelligence data in STIX™v2.0 [4] format,
- DRM.owl for dynamic risk data.

Both are connected as DRM.owl directly imports and extends STIX2.owl.

OWL DL (description logic) is designed to provide the maximum expressiveness as possible while retaining computational completeness (either Φ or $\neg\Phi$ holds), decidability (there is an effective procedure to determine whether Φ is derivable or not), and the availability of practical reasoning algorithms.

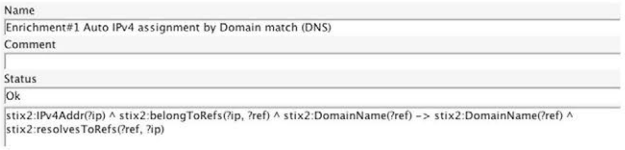
STI2.owl is a contribution to help STIX™v2.0 [4] to solve their current problems with regard to the representation of more complex concepts like TTP (techniques, tactics and procedures) [6], campaigns [7] or incidents [8] but also to make automation easier by using semantic reasoners. We followed all OASIS open standard specification of its version 2.0 translating the whole standard, that is to say, all requirements and restrictions from the STIX™specifications are now ontology classes, property objects, data types and axioms in OWL format.

DRM.owl imports STIX2.owl leveraging CTI into a more comprehensive and meaningful DRA/DRM architecture.

Taxonomies and domains are fully integrated: concepts are related between themselves, and relationships are formally established between all types in order to help us to solve initial challenges but also providing us the capability to use a reasoner (this time we used Pellet incremental reasoner) [14]. Data types are formally defined as well.

We then have a graph and meaningful data model to connect a specific IOC, threat or security event to an asset. We also have the possibility to calculate associated risks around each service, based on dependencies, as seen in Fig. 1.

The framework presented is a formal model which enables the representation of any CTI or DRM context despite its complexity and previous limitations like the ones still not solved in STIX™XML and JSON format [6–8] evolving threat intelligence data until today into real TTP meaningful patterns and representations (graphs beyond connected IOC).



Name

Enrichment#1 Auto IPv4 assignment by Domain match (DNS)

Comment

Status

Ok

```
stix2:IPv4Addr(?ip) ^ stix2:belongToRefs(?ip, ?ref) ^ stix2:DomainName(?ref) -> stix2:DomainName(?ref) ^
stix2:resolvesToRefs(?ref, ?ip)
```

Fig. 2 SWRL DNS enrichment

Due to our contribution of a complete DRM framework leveraging a complete version of STIX™v2.0 ontology, any STIX™related data can be parsed into our ontology.

In our work, all the relevant data for our use case at all levels (operational, tactic, strategic) were parsed into our DRM ontology as a central data storage. All SWRL rules handle or evaluate these data. On the other hand, we limited to STIX™the different types of data to be collected as future sensors will probably provide data in a format compatible with this taxonomy. We have then focused our contribution to the processing of data but not about its collection.

2.2.2 Semantic business logic (reasoning) model

Now that all data (threat intelligence and risk data) are based on OWL and relationships are formally established, we propose to use SWRL (Semantic Web Rule Language) which will enable us to express rules as well as logic combining OWL DL, OWL Lite and RuleML (Rule Markup Language).

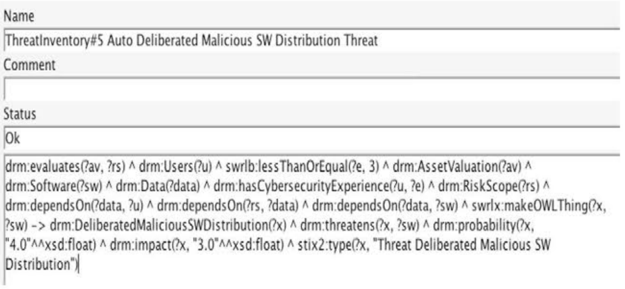
Rules are of the form of an implication between an antecedent (body) and consequent (head). The intended meaning can be read as: Whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold.

We will use SWRL to create semantic algorithms and rules that will use semantic data to provide value added and to cover all use cases defined above, like:

(a) Enrichment of threat information context, for example, by using the SWRL rule of Fig. 2, we can represent a simple DNS data enrichment to make relationships between IP and domain concepts. By knowing an IP address belonging to a DomainName, we create a reverse relationship, that is, the DomainName to IP is the "inverseOf" IP to DomainName.

The rule has an antecedent (body) indicating that all IP addresses in IPv4 format belonging to a specific domain name will then create an inverse relationship in the consequent (head), that is to say, the same domain names will then resolve back to those IP addresses. This is a DNS versus reverse DNS behavior.

As we implemented the whole STIX™v2.0 [4] taxonomy, the rule uses the unambiguous concept (classes) "IPv4Addr" to differentiate from "IPv6Addr", also in STIX™taxonomy the property named belongToRef and the domain name is the concept DomainName (without spaces).



Name

ThreatInventory#5 Auto Deliberated Malicious SW Distribution Threat

Comment

Status

Ok

```
drm:evaluates(?av, ?rs) ^ drm:Users(?u) ^ swrlb:lessThanOrEqual(?e, 3) ^ drm:AssetValuation(?av) ^
drm:Software(?sw) ^ drm:Data(?data) ^ drm:hasCybersecurityExperience(?u, ?e) ^ drm:RiskScope(?rs) ^
drm:dependsOn(?data, ?u) ^ drm:dependsOn(?rs, ?data) ^ drm:dependsOn(?data, ?sw) ^ swrlx:makeOWLThing(?x,
?sw) -> drm:DeliberatedMaliciousSWDistribution(?x) ^ drm:threatens(?x, ?sw) ^ drm:probability(?x,
"4.0"^^xsd:float) ^ drm:impact(?x, "3.0"^^xsd:float) ^ stix2:type(?x, "Threat Deliberated Malicious SW
Distribution")
```

Fig. 3 SWRL rule to make automatic inventory of deliberated malicious SW distribution threat inventory when a service in scope of our risk management framework depends on data and later on SW (OS, Browser, AdobeFlashPlugin) but also on HW (Personal Computers) used by internal users with little cybersecurity experience <= 3

Once this SWRL rule is created and imported as an axiom into the ontology, the reasoner, when active, will make automatic reasoning by filling and enriching all new domain names or IP addresses if there is enough info related to the antecedent to execute the consequent (head).

This is a very simple example, but we can also create more complex enrichment rules by using SWRL as our semantic business logic model.

Anyway, SWRL rules are quite easy to understand and to create by non-experience users. A 4-h briefing about SWRL and the namespace domains used (e.g., OWL ontologies: classes, properties and data types of both domains) should be enough to start creating business oriented SWRL rules. Until today, complex rules had to be created by programmers with high SW programming skills [33]. We propose SWRL rules to solve this problem and to enable any person (operator (operational), manager (tactical) or director (strategic)) to create its own rules.

(b) Enrichment of risk assessment and risk management context. We are now able to work in all phases dynamically; for example, by using the SWRL rule presented in Fig. 3, we automatically make an automated threat inventory of "Deliberated Malicious SW Distribution Threats" type.

This will make an automatic inventory of threats by a more advanced pattern which is useful to risk owners if it can be automatically detected by reasoners. The algorithm is taking into consideration specific patterns when potential deliberated malicious windows executables are dropped during Web surfing traffic. However, part of this behavior could happen with licit content, and strange Javascript redirections together with lack of cybersecurity expertise by the end user will be also considered in the business logic of the algorithm as shown inside the SWRL rule. That is to say, the same algorithm will not create automatic threats with that level of probability if the end user navigating is an expert (the system understands that the probability of a fake installer to be executed by an expert is residual).

The algorithm antecedent of the figure will take into account the situation when an essential service depends on specific data which also depends on SW used by an internal (corporate) user who has access to it at the same time he/she has low cybersecurity experience (cybersecurity experience $<= 3$). The threat is then recognized as a potential risk in specific situations. Situations can be of any type as we count with enough expressivity to define patterns and behaviors in our framework.

Taking into account the level of cybersecurity experience of an end user, it is clear that it will influence the probability to execute (or not) a specific malicious installer or dropper. The risk level for this type of threats (malicious SW distribution) will be different if the end user has different levels of expertise (the impact could be the same but not the probability of occurrence).

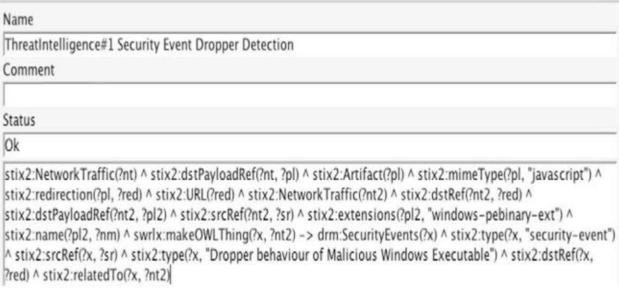
In this rule, we can see that most concepts used are defined into the drm ontology (see drm: prefix before the concept used, the prefix defines the ontology where is defined). We also use built-in features like swrlb and swrlx prefixes used for different purposes, like comparison features, math calculations or even to create new individual instances. In the rule, we also use a class property defined in stix2 (see stix2 prefix in the rule). At the end, we have an integrated framework where either stix2 or drm can be used in the same business logic (algorithm) together with SWRL built-in functions.

In Fig. 4, we implemented a rule to detect real malicious events related to this type of risk. This specific SWRL rule checks redirection after redirection until an executable file is dropped, all events following STIX™standard. SWRL permits enhanced rules to describe a specific semantic pattern in the network traffic without having knowledge about specific IOC involved in the attack. The SWRL rule is then IOC agnostic with regard to domain name, IP or hashes.

It is an effective cyber threat intelligence rule that can also be shared between mates without the risk of exposing your corporate data. This is one example of how two or more organizations can share intelligence and build trust without sharing real data or IOCs between them. By using ontologies (as a formal model) and standards, rules can be easily loaded into new organizations upgrading their detection capabilities rapidly. This is possible when the reasoner uses and understands the same language which is a real contribution to the state of the art.

This type of service/project will likely have more risk related to this type of threats when used by non expert users versus when they are used only by cybersecurity expert users. Risks are guessed automatically taking all this granularity into consideration.

c) Automation of risk management like the automation of risk level classification, as a previous step to decision making. The SWRL rule of Fig. 5 makes the automatic classification of risks belonging to what it is defined by the organization as high risks along the time.



Name

ThreatIntelligence#1 Security Event Dropper Detection

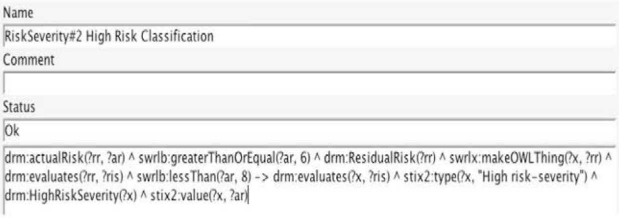
Comment

Status

Ok

```
stix2:NetworkTraffic(?nt) ^ stix2:dstPayloadRef(?nt, ?pl) ^ stix2:Artifact(?pl) ^ stix2:mimeType(?pl, "javascript") ^
stix2:redirection(?pl, ?red) ^ stix2:URL(?red) ^ stix2:NetworkTraffic(?nt2) ^ stix2:dstRef(?nt2, ?red) ^
stix2:dstPayloadRef(?nt2, ?pl2) ^ stix2:srcRef(?nt2, ?sr) ^ stix2:extensions(?pl2, "windows-pebinary-ext") ^
stix2:name(?pl2, ?nm) ^ swrlx:makeOWLThing(?x, ?nt2) -> drm:SecurityEvents(?x) ^ stix2:type(?x, "security-event")
^ stix2:srcRef(?x, ?sr) ^ stix2:type(?x, "Dropper behaviour of Malicious Windows Executable") ^ stix2:dstRef(?x,
?red) ^ stix2:relatedTo(?x, ?nt2)
```

Fig. 4 Pattern-based SWRL rule to detect a security event that will be triggered by a dropper executable delivered automatically after a network traffic redirection of an injected Javascript takes place. It has not specific IOC, and all individuals are variables



Name

RiskSeverity#2 High Risk Classification

Comment

Status

Ok

```
drm:actualRisk(?rr, ?ar) ^ swrlb:greaterThanOrEqual(?ar, 6) ^ drm:ResidualRisk(?rr) ^ swrlx:makeOWLThing(?x, ?rr) ^
drm:evaluates(?rr, ?ris) ^ swrlb:lessThan(?ar, 8) -> drm:evaluates(?x, ?ris) ^ stix2:type(?x, "High risk-severity") ^
drm:HighRiskSeverity(?x) ^ stix2:value(?x, ?ar)
```

Fig. 5 SWRL rule for auto classification of high severity risks

By using SWRL, we are able to easily implement the business logic of our threat and risk domains. By using the integrated ontologies, our rules based on SWRL could handle any data from any domain within the same algorithm in either antecedent or consequent. It fulfills the needed expressivity. Once new data came in, our reasoner will make dynamic, automatic and semantic reasoning. This could help CTI or DRA/DRM domains, for example making automatic threat inventory explained in Fig. 3 or even more complex reasoning like triggering an alert in the services/application layer when a TTP occurs like the pattern explained in Fig. 4.

As an example of potential efficiency, nowadays, to mitigate one threat, there are always multiple rules and filters if they are based on IOCs. By using our model, a threat's TTP can be described into a SWRL rule to have the same effect that multiple IOC-based rules (matching IOC). As a result, we can be more efficient as we can cover and filter multiple mutations of the same threat in a single pattern-based rule. From an operator point of view, less rules and a reasoner are better team mates than multiple IOC-based rules to block one threat.

2.2.3 Semantic services/application model

Here we provide services and applications for both domains. We can also give answer to more complex scenarios and use cases in the services/application layer by using SWRL expressivity as well.

As an example, we propose a rule for CTI security event detection. By using a more sophisticated SWRL rule, we can represent a security event detection based on a TTP (which now can be represented without semantic limitations [5–8]) which business logic is based on a specific network traffic pattern. The pattern, once it matches, will increase dynamically the risk level of those risks of type deliberated malicious SW distribution.

The following SWRL alert will analyze and represent a TTP when an injected Javascript is loaded after visiting a compromised Web site, and it redirects an internal corporate user to a different URL where a Windows executable file (*.exe) is automatically retrieved. The SWRL rule in Fig. 4 creates an individual instance of class SecurityEvent (meaning a security alert). The creation is done at "swrlx:makeOWLThing" using variable x (?x means). After that, a drm:SecurityEvents(?x) is defining the class of the individual instance of that variable. This rule will analyze two different network traffics: One depends on the other: first connection made a redirection into another URL which destination is dropping an executable artifact as dstPayloadRef. All this naming convention came from STIX™standard by OASIS that we implemented in OWL as proposed in STIX™white paper as potential future research and implementation [5].

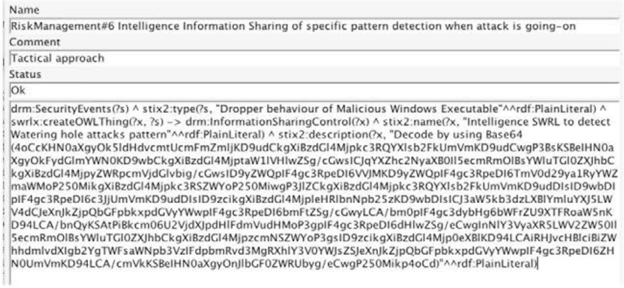
We are also able to solve challenges like security incident detection and prevention, for example based on security events like the example described in Fig. 4.

Other types of CTI challenges like incident triage by severity (incident handling) and information sharing control (why, what, when and how) will have their corresponding SWRL rules, but also they will be meaningful (SWRL with OWL DL graph-based data) also for reasoners.

Information sharing application could also be better guaranteed by implementing STIX™v2.0 [4] Marking definition, for example, to control how data can be used and shared. For example, implementing TLP as a marking definition restriction where data may be shared with the restriction that it must not be re-shared, or that it must be encrypted at rest.

We implemented a concrete example of a **risk management action at tactical level** and other example of a **risk management action at strategic level** to demonstrate how our framework enables the connection from CTI into a complete DRM, at all levels.

With regard to the **tactical level**, we propose to share risk intelligence information as a preventive action once we are receiving a specific attack in one office to protect other remote offices and/or partners. In our case, we will create an instance of class "Information Sharing Control" dynamically once this situation is happening. In our case, we have two options either to share details about IoC or to send the intelligence rule that is how receivers can detect by themselves the same incident if using our formal model framework. The SWRL rule that detects the pattern attack will be shared itself inside another SWRL information sharing rule dynamically once the attack is being received. This tactic will facilitate efficient and dynamic detection and protection during a campaign as part of our corporate tactics. The new rule will not include IoC but the rule itself (algorithm) in order to demonstrate other of our contributions (risk intelligence sharing rules without IOC/data). This changes the paradigm of information sharing until today. By using a formal model and ontologies, rules can be applied as plug and play if using the same framework.



Name

RiskManagement#6 Intelligence Information Sharing of specific pattern detection when attack is going-on

Comment

Tactical approach

Status

Ok

```
drm:SecurityEvents(?s) ^ stix2:type(?s, "Dropper behaviour of Malicious Windows Executable"^^rdf:PlainLiteral) ^
swrlx:createOWLThing(?x, ?s) -> drm:InformationSharingControl(?x) ^ stix2:name(?x, "Intelligence SWRL to detect
Watering hole attacks pattern"^^rdf:PlainLiteral) ^ stix2:description(?x, "Decode by using Base64
(4oCcKHN0aXgyOk5ldHdvcmtUcmFmZmljKD9udCkgXiBzdGl4Mjpkc3RQYXlsb2FkUmVmKD9udCwgP3BsKSBeIHN0a
XgyOkFydGlmYWN0KD9wbCkgXiBzdGl4MjptaW1lVHlwZSg/cGwsICJqYXZhc2NyaXB0IiI5ecmRmOlBsYWluTGl0ZXJhbC
kgXiBzdGl4MjpyZWRpcmVjdGlvbig/cGwsID9yZWQpIF4gc3RpeDI6VVJMKD9yZWQpIF4gc3RpeDI6TmV0d29ya1RyYWZ
maWMoP250MikgXiBzdGl4Mjpkc3RSZWYoP250MiwgP3JlZCkgXiBzdGl4Mjpkc3RQYXlsb2FkUmVmKD9udDIsID9wbDI
pIF4gc3RpeDI6c3JjUmVmKD9udDIsID9zcikgXiBzdGl4MjpleHRlbnNpb25zKD9wbDIsICJ3aW5kb3dzLXBlYmluYXJ5LW
V4dCJeXnJkZjpQbGFpbkxpdGVyYWwpIF4gc3RpeDI6bmFtZSg/cGwyLCA/bm0pIF4gc3dybHg6bWFrZU9XTFRoaW5nK
D94LCA/bnQyKSAtPiBkcm06U2VjdXJpdHlFdmVudHMoP3gpIF4gc3RpeDI6dHlwZSg/eCwgInNlY3VyaXR5LWV2ZW50Il
5ecmRmOlBsYWluTGl0ZXJhbCkgXiBzdGl4MjpzcmNSZWYoP3gsID9zcikgXiBzdGl4Mjp0eXBlKD94LCAiRHJvcHBlciBiZW
hhdmlvdXIgb2YgTWFsaWNpb3VzIFdpbmRvd3MgRXhlY3V0YWJsZSJeXnJkZjpQbGFpbkxpdGVyYWwpIF4gc3RpeDI6ZH
N0UmVmKD94LCA/cmVkKSBeIHN0aXgyOnJlbGF0ZWRUbyg/eCwgP250Mikp4oCd)"^^rdf:PlainLiteral)
```

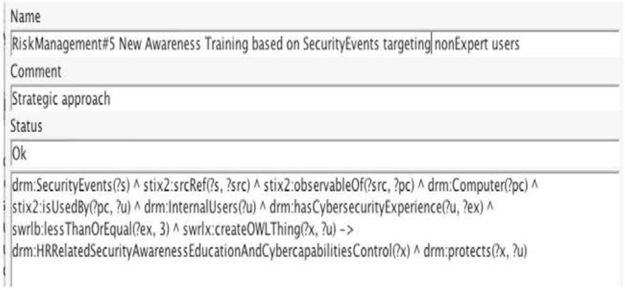
Fig. 6 SWRL rule to implement a specific **risk management tactic**: Once a specific type of attack is detected, the detection algorithm (not the data or IOC) will be shared with the rest of the offices or partners to transfer knowledge (to improve their own detection capabilities). It enables the detection of the same attack even if using modified IOCs but the same TTP

In detail, once we have a detection of a security event of type "Dropper behavior of Malicious Windows Executable," we will create an instance of a ISO27K control sharing class named "Information Sharing Control" to share it with our mates. The new instance will include inside the related algorithm for detection (SWRL detection rule) in order to transfer knowledge about how to detect it. It will help others to better prevent similar incidents with different IOCs but the same TTP by improving their detection capabilities beyond specific IOCs.

We decided to encode the algorithm or SWRL rule being shared into base64 (it could also be ciphered if needed). The rule to share is the same as in Fig. 4.

The SWRL rule at tactic risk management level that will create an information sharing control to manage intelligence sharing is the one in Fig. 6

With regard to the **strategic level**, the same security incident will trigger specific "Awareness trainings" as a dynamic RM decision proposed by the reasoner using NRT and the claimed expressiveness. That is to say, the reasoner proposes dynamically specific awareness trainings only to those employees which are receiving real and specific security threats (security event instances), and at the same time, they still do not count with enough experience in cybersecurity (rating equal or below 3 out of 5). This allows the organization



Name

RiskManagement#5 New Awareness Training based on SecurityEvents targeting nonExpert users

Comment

Strategic approach

Status

Ok

```
drm:SecurityEvents(?s) ^ stix2:srcRef(?s, ?src) ^ stix2:observableOf(?src, ?pc) ^ drm:Computer(?pc) ^
stix2:isUsedBy(?pc, ?u) ^ drm:InternalUsers(?u) ^ drm:hasCybersecurityExperience(?u, ?ex) ^
swrlb:lessThanOrEqual(?ex, 3) ^ swrlx:createOWLThing(?x, ?u) ->
drm:HRRelatedSecurityAwarenessEducationAndCybercapabilitiesControl(?x) ^ drm:protects(?x, ?u)
```

Fig. 7 SWRL rule to propose a specific and detailed **risk management strategy**: Once a specific type of attack is detected, the reasoner will propose a specific awareness training to those specific users which are affected by the threat, and at the same time, they do not have enough cybersecurity experience

to launch more efficient trainings with so much granularity and dynamically. In this case, the SWRL rule is shown in Fig. 7.

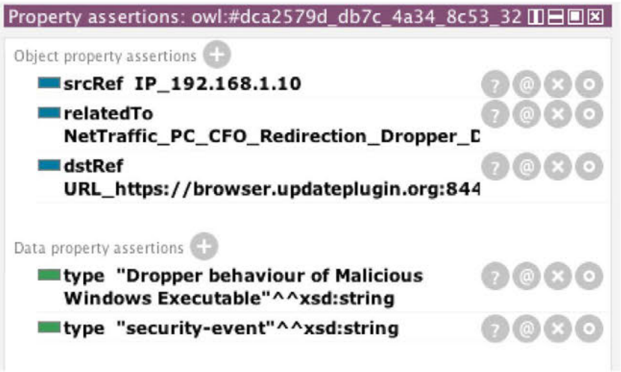
In detail, the reasoner instances new strategic safeguards or controls dynamically of class "HR Related Security Awareness Education and Cyber capabilities control" indicating the need to protect (property drm:protects) each specific user (class drm:InternalUsers) associated with real-time threats. We also consider in the rule that only non-experience users in cybersecurity will be eligible for the training (rating equal or below 3 out of 5, for that we use drm:hasCybersecurityExperience property AND swrlb: lessThanOrEqual operator). Additionally, we can create content-oriented SWRL rules to make specific types of training depending on the threats (e.g., social engineering, Web surfing, etc.). With this expressiveness, there could be customized trainings proposed by the reasoner based on real but specific classification of threats.

Another interesting benefit is that we could leverage sensors into more intelligence sensors by letting them access structured OWL data as well as using SWRL rules or even using SQWRL (Semantic Query Web Rule Language that will be explained in the next section) to query the formal model by using a syntax close to SQL. There would be high benefits if all the topology is using the same data model standards and ontologies, even sensors.

2.2.4 Semantic visualization model

Because our data, business logic and services/applications are based on semantic OWL data, SWRL and reasoners; our visualization model could benefit from enriched semantic graphs. We have different options by using Protegé tool created by Stanford.

- OntoGraph plug-in to render interactive graphs of our integrated formal model (CTI + DRA/DRM classes, properties and data types). It also includes individual instances (threat intelligence and risk-related data) as shown in Fig. 1. Equivalent plug-ins could be VOWL and OntoViz.



Property assertions: owl:#dca2579d_db7c_4a34_8c53_32

Object property assertions

srcRef IP_192.168.1.10

relatedTo NetTraffic_PC_CFO_Redirection_Dropper_D

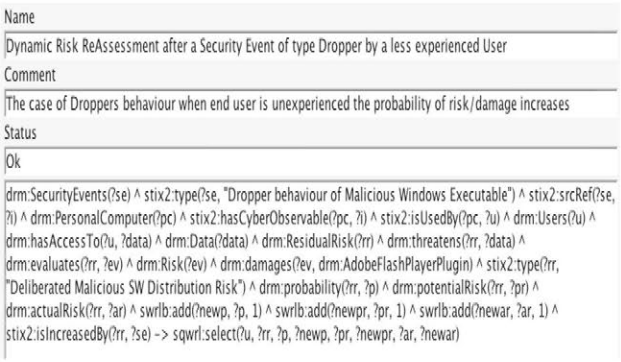
dstRef URL_https://browser.updateplugin.org:844

Data property assertions

type "Dropper behaviour of Malicious Windows Executable"^^xsd:string

type "security-event"^^xsd:string

Fig. 8 Screenshot in Protegé tool of the security event created by the SWRL rule seen in Fig. 4 when our CFO browser has been redirected into a fake Flash (exe) installer. The pattern, redirections + dropper are part of the TTP definition



Name

Dynamic Risk ReAssessment after a Security Event of type Dropper by a less experienced User

Comment

The case of Droppers behaviour when end user is unexperienced the probability of risk/damage increases

Status

Ok

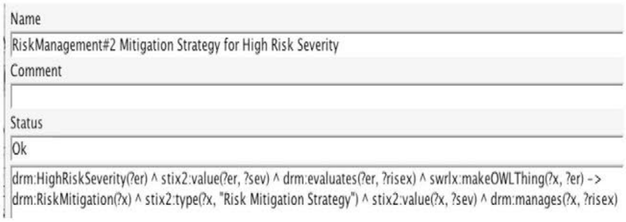
```
drm:SecurityEvents(?se) ^ stix2:type(?se, "Dropper behaviour of Malicious Windows Executable") ^ stix2:srcRef(?se,
?i) ^ drm:PersonalComputer(?pc) ^ stix2:hasCyberObservable(?pc, ?i) ^ stix2:isUsedBy(?pc, ?u) ^ drm:Users(?u) ^
drm:hasAccessTo(?u, ?data) ^ drm:Data(?data) ^ drm:ResidualRisk(?rr) ^ drm:threatens(?rr, ?data) ^
drm:evaluates(?rr, ?ev) ^ drm:Risk(?ev) ^ drm:damages(?ev, drm:AdobeFlashPlayerPlugin) ^ stix2:type(?rr,
"Deliberated Malicious SW Distribution Risk") ^ drm:probability(?rr, ?p) ^ drm:potentialRisk(?rr, ?pr) ^
drm:actualRisk(?rr, ?ar) ^ swrlb:add(?newp, ?p, 1) ^ swrlb:add(?newpr, ?pr, 1) ^ swrlb:add(?newar, ?ar, 1) ^
stix2:isIncreasedBy(?rr, ?se) -> sqwrl:select(?u, ?rr, ?p, ?newp, ?pr, ?newpr, ?ar, ?newar)
```

Fig. 9 SQWRL rule for risk re-calculations after a security event is detected

- Protege standard interface where object and data property assertions can be read (Fig. 8). - SQWRL query language to query our data model or instances. An example of query is shown in Fig. 9 and results of that query are shown in Fig. 13. When consequent belongs to a query, "sqwrl:select" is used.

With regard to risk assessment and risk management processes, our model is able to accommodate the specificities of the selected organization (the National CSIRT) whose risk assessment and risk management model is described in Sect. 2.1.2.

CTI data is already integrated with risk assessment and risk management data. Some examples of this NRT (near real time) integration can be shown in Fig. 4 where prefix "stix2" represent our new ontology implementation of STIX™v2.0 [4] data and prefix "drm" represent our new ontology implementation of DRA/DRM (dynamic risk assessment and dynamic risk management) as well as their relationships, rules and axioms.



Name

RiskManagement#2 Mitigation Strategy for High Risk Severity

Comment

Status

Ok

```
drm:HighRiskSeverity(?er) ^ stix2:value(?er, ?sev) ^ drm:evaluates(?er, ?risex) ^ swrlx:makeOWLThing(?x, ?er) ->
drm:RiskMitigation(?x) ^ stix2:type(?x, "Risk Mitigation Strategy") ^ stix2:value(?x, ?sev) ^ drm:manages(?x, ?risex)
```

Fig. 10 SWRL rule used in risk management for automatic classification. Once a high severity risk is detected, it is automatically classified for a mitigation strategy because the risk is not acceptable for the organization as is

Another example of a dynamic risk management process is the automation of the better strategy depending on the risk and its severity. The SWRL rule for this can be shown in Fig. 10.

As seen in Figs. 9 and 13, our dynamic risk assessment approach would also allow risks to be updated along the time due to special security events. The one shown in Fig. 4 is an example of an event that will increase the risk once it is detected. Risks can be increased by changes on two factors or variables: impact or probability. In this case, the probability will change as something strange is close to become an incident related to that type of risk.

As seen in Fig. 6, specific algorithms have been implemented to help the reasoner to propose specific **tactics**. In our case, once a security event is detected of a specific type, the reasoner will create an instance to share the detection algorithm as a risk intelligence sharing practice. The algorithm encoded in base64 will be shared to other offices or partners once an attack has been detected in our office. We are then transferring knowledge and detection capabilities by transferring the algorithm or SWRL rule inside another SWRL rule. They will be able to detect the same pattern once the SWRL rule shared is loaded in their systems even if attack IoC has changed. The only condition is that the TTP pattern remains the same as our rule is an algorithm to detect a TTP pattern. There will be different SWRL rules to detect different TTP patterns. The instance created automatically (prefix owl followed by a hash means automatically generated instances) can be seen in Fig. 11 together with the contents of the instance; in this case, it includes the base64 encoding version of rule seen in Fig. 4.

At the same time, in Fig. 7, specific algorithms have been implemented to help the reasoner to propose specific **strategies** about HR awareness trainings. In our case, once a security event is detected of a specific type, the reasoner will create an instance of recommended awareness training to specific users which are receiving real-time attacks and can be vulnerable to those attacks due to their cybersecurity experience. As seen in Fig. 12, a new instance of the suggested control class (HR Related Security Awareness Education and Cyber capabilities control) is created dynamically by the SWRL rule of Fig. 7 considering the specific internal user that is recommended for the training.

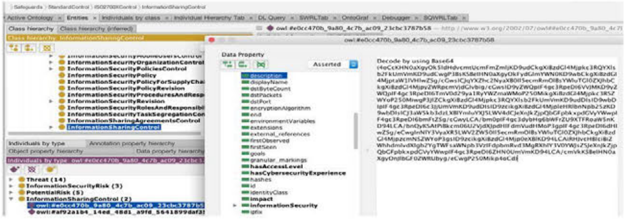

Safeguards › StandardControl › ISO2700XControl › InformationSharingControl
Active Ontology × Entities × Individuals by class × Individual Hierarchy Tab × DL Query × SWRLTab × OntoGraf × Debugger × SQWRLTab ×
Class hierarchy Class hierarchy (inferred)
owl:#e0cc470b_9a80_4c7b_ac09_23cbc3787b58 — http://www.w3.org/2002/07/owl##e0cc470b_9a80_4c7
Class hierarchy: InformationSharingControl
owl:#e0cc470b_9a80_4c7b_ac09_23cbc3787b58
InformationSecurityOrganizationControl
InformationSecurityPoliciesControl
InformationSecurityPolicy
InformationSecurityPolicyForSupplyCha
InformationSecurityPolicyRevision
InformationSecurityProceduresAndResp
InformationSecurityRevision
InformationSecurityRolesAndResponsibi
InformationSecurityTaskSegregationCon
InformationSharingAgreementsControl
InformationSharingControl
Data Property
Asserted
description
displayName
dstByteCount
dstPackets
dstPort
encryptionAlgorithm
end
environmentVariables
extensions
external_references
firstObserved
firstSeen
goals
granular_markings
hasAccessLevel
hasCybersecurityExperience
hashes
id
identityClass
impact
informationSecurity
ipfix
Decode by using Base64
Individuals by type Annotation property hierarchy
Object property hierarchy Data property hierarchy
Individuals by type: owl:#e0cc470b_9a80_4c7b_ac09_23cbc3
Threat (14)
InformationSecurityRisk (3)
PotentialRisk (5)
InformationSharingControl (2)
owl:#e0cc470b_9a80_4c7b_ac09_23cbc3787b5
owl:#af92a1b4_14ed_48d1_a9fd_5641899daf3

Fig. 11 Screenshot in Protegé tool of the instance created dynamically of class information sharing control which at the same time is including the encoded version of the SWRL attack detection TTP

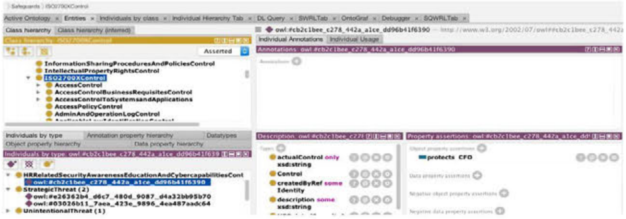

Safeguards › ISO2700XControl
Active Ontology × Entities × Individuals by class × Individual Hierarchy Tab × DL Query × SWRLTab × OntoGraf × Debugger × SQWRLTab ×
Class hierarchy Class hierarchy (inferred)
owl:#cb2c1bee_c278_442a_a1ce_dd96b41f6390 — http://www.w3.org/2002/07/owl##cb2c1bee_c278_442a
Class hierarchy: ISO2700XControl
Asserted
Individual Annotations Individual Usage
Annotations: owl:#cb2c1bee_c278_442a_a1ce_dd96b41f6390
Annotations
InformationSharingProceduresAndPoliciesControl
IntellectualPropertyRightsControl
ISO2700XControl
AccessControl
AccessControlBusinessRequisitesControl
AccessControlToSystemsandApplications
AccessPolicyControl
AdminAndOperationLogControl
Individuals by type Annotation property hierarchy Datatypes
Object property hierarchy Data property hierarchy
Individuals by type: owl:#cb2c1bee_c278_442a_a1ce_dd96b41f639
HRRelatedSecurityAwarenessEducationAndCybercapabilitiesCont
owl:#cb2c1bee_c278_442a_a1ce_dd96b41f6390
StrategicThreat (2)
owl:#e26362b4_d6c7_480d_9087_d4a32bb95b70
owl:#03026b11_7aea_423e_9896_4ea487aadc64
UnintentionalThreat (1)
Description: owl:#cb2c1bee_c27
Types
actualControl only xsd:string
Control
createdByRef some Identity
description some xsd:string
Property assertions: owl:#cb2c1bee_c278_442a_a1ce_dd
Object property assertions
protects CFO
Data property assertions
Negative object property assertions
Negative data property assertions

Fig. 12 Screenshot in Protegé tool of the instance created dynamically of class HR Related Security Awareness Education and Cyber capabilities control as recommended training by the reasoner due to real-time security events

2.2.5 Feasibility of a new dynamic risk equation

By using our framework, we can then improve the selected organization's equation 1 to leverage risk assessment into something dynamic (depending on time t and security events) and affordable with enough flexibility and granularity like:

$$R_{it} = P_{it} + I_{it} - \Sigma(C1_{it}, C2_{it}, \ldots, CN_{it}) \quad (2)$$

$$R_t = \Sigma R_{it} \quad (3)$$

being:

$$P_{it} = P_{it-1} + \Sigma(EP1_{it}, EP2_{it}, \ldots, EPN_{it}) \quad (4)$$

$$I_{it} = I_{it-1} + \Sigma(EI1_{it}, EI2_{it}, \ldots, EIN_{it}) \quad (5)$$

where:

- P_{it} is the probability of threat i at time t,
- I_{it} is the impact of threat i at time t,
- CN_{it} is the decreasing value of the impact (severity) I or probability P of threat i at time t due to countermeasure N,
- EPN_{it} is the increasing value of the probability P of threat i at time t due to the security event N,
- EIN_{it} is the increasing value of the impact (severity) I of threat i at time t due to the security event N,
- R_{it} is the residual risk associated with threat i, at time t,

- ΣR_{i_t} is the sum of all type of residual risks associated with threats $i = 1, \ldots, Z$, at time t,
- R_t is the total residual risk at time t of all type of threats $i = 1, \ldots, Z$,

We are finally able to implement the dynamic risk assessment equation 2 by using our model. We use SWRL rules to decrease values when a safeguard (CN) mitigates that risk. There are also rules to increase values when a security event (EPN or EIN) increases the same risk.

In our case:

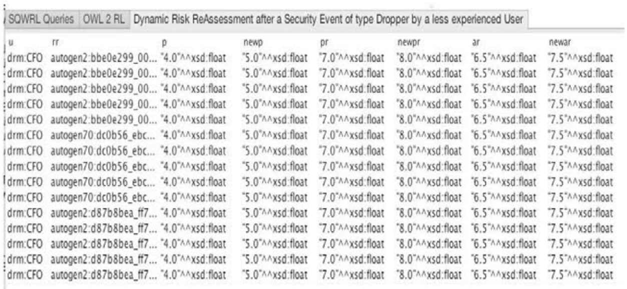
$$R_{i_{t=1}} = P_{i_{t=1}} + I_{i_{t=1}} - \Sigma(C1_{i_{t=1}}, C2_{i_{t=1}}), \tag{6}$$

$$P_{i_{t=1}} = P_{i_{t=0}} + 1 \tag{7}$$

$$I_{i_{t=1}} = I_{i_{t=0}} + 0 \tag{8}$$

where:

- i is the "Threat of Deliberated Malicious SW Distribution,"
- $P_{i_{t=1}}$ is the probability of that threat at time $t = 1$, that is "newp" of Fig. 13,
- $I_{i_{t=1}}$ is the impact of that threat at time $t = 1$, which is equal to the impact at $t = 0$ because the recent security event is only updating the probability,
- $C1_{i_{t=1}}$ is the decreasing value of the Impact (severity) i or probability P of that threat i at time $t = 1$ due to countermeasure 1 which is the "Antivirus" (reducing 0.5 float),
- $C2_{i_{t=1}}$ is the decreasing value of the impact (Severity) I or Probability P of that threat i at time $t = 1$ due to countermeasure 2 which is the "LDAP" (reducing 0.0 float as it is not considered a countermeasure for this type of threats),
- $EPN_{i_{t=1}}$ which is equal to 1 because it is the increasing value of the probability P of that threat i at time $t = 1$ due to the security event $N = 1$; in our case, it is 1.0 float once the security event of a dropper has been detected,
- $EIN_{i_{t=1}}$ which is equal to 0 because it is the increasing value of the impact (severity) I of that threat i at time t due to the security event $N = 1$; in our case, it is 0.0 float, and the security event updates the probability not the impact variable.
- $R_{i_{t=1}}$ is the residual risk associated with that threat i, at time $t = 1$, it is "newar" of Fig. 13,


SQWRL Queries
OWL 2 RL
Dynamic Risk ReAssessment after a Security Event of type Dropper by a less experienced User

u	rr	p	newp	pr	newpr	ar	newar
drm:CFO	autogen2:bbe0e299_00...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen2:bbe0e299_00...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen2:bbe0e299_00...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen2:bbe0e299_00...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen2:bbe0e299_00...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen70:dc0b56_ebc...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen70:dc0b56_ebc...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen70:dc0b56_ebc...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen70:dc0b56_ebc...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen70:dc0b56_ebc...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen2:d87b8bea_ff7...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen2:d87b8bea_ff7...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen2:d87b8bea_ff7...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen2:d87b8bea_ff7...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen2:d87b8bea_ff7...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float

Fig. 13 Results of querying rule at Fig. 9

Figure 13 shows how the probability changes after a suspicious pattern is detected as it increases the probability of a specific threat type.

New values are calculated, and they can be seen in Fig. 13 after a dropper has been detected inside network traffic of the CFO User (see Fig. 9 to check its related SQWRL query). As an example:

- p=previous probability → newp=new probability ,
- pr=previous potential risk → newpr = new potential risk and
- ar=previous actual risk → newar = new actual risk.

Risks assessments will be updated dynamically, so does its risk management classification and their recommended treatment.

Until today, our selected organization is re-assessing risks once a year. Risks considered by our organization are also related to threats identified by hand by risk owners. Those are usually reflected in an Excel file or any other static file. By using our proposal, we are providing to our organization the benefit to leverage their CTI into a dynamic risk management framework. The proposal is based on standards [4,10,11], and then, any other organization who will be working under these standards will have the same benefit as well.

In our case, a risk that had been identified as a medium-level risk by the organization is dynamically re-classified under high-level risk once a threat is detected that will potentially be affecting a classified data. The reasoner will provide reasoning evidences for that, in our case, a specific user (CFO) with low-level cybersecurity expertise but, with high-level access to that classified data, is under attack. As this user has inherently more probability to accept this malicious fake installer, the dynamic risk framework will trigger a re-classification of the risk with that granularity (details) in that specific moment, before it becomes a real incident. It is then a proactive and preventive response with again, different options or alternatives.

Semantics would allow a pseudo-automated response execution where complex and/or special actions would require a balanced human supervision and interaction (HMI). This is our recommendation for all decision-making actions related to cybersecurity domain.

As a result, here we propose some examples of different types of automated responses within a dynamic risk management framework:

- Security policy changes (e.g., blocking traffic, increase password robustness, etc.).
- Patching remediation and reprioritization queues.
- Risk transfer to a third party.
- New rules or signatures for networking devices.
- Risk Information Sharing.
- Alert notification.
- Launch honeypot and counterintelligence (deception) measures.
- Distributed topology based on software-defined networks (as a joint) response (e.g., sensing, monitoring, or even taking offensive actions).
- Risk exposure escalation to management including visualization to support decision making based on historical action and responses.
- Etc.

3 Use case: watering hole attack

We selected a real attack from August 2017 that affected a popular middle east news site. The attacks of type watering hole are usually very sophisticated attacks as well as they are very difficult to detect.

This is the main reason for us to select this specific attack in this work. There is a difficulty to detect this type of attacks if our detection techniques are only based on knowns IOCs (indicators of compromise), it forces us to deploy new detection techniques based on behavioral patterns to detect and fight the still unknown and dangerous attacks. Our proposal is based on standards, and it adds so much expressiveness by using ontologies, so it will allow us to create any pattern detection rule in SWRL beyond watering hole attacks. It will help us to detect any other emerging unknown attack based on patterns. At the same time, all risk information will be updated dynamically after a dynamic re-assessment. It will be presented accordingly to tactical and strategic management, but it will also may include any type of automatic responses. Here we propose a pseudo-automatic response, at least for offensive ones, as we consider that a human should take the ultimate decision in this type of attacks.

A watering hole attack is a computer attack strategy, in which the victim is a particular group (organization, industry, or region). In this attack, the attacker guesses or observes which Web sites the group often use and infect one or more of them with malware. Eventually, some member of the targeted group becomes infected.

Hacks looking for specific information may only attack users coming from a specific IP address (same request from different IPs might result in different responses), so malware will be only delivered to our victims. On the other hand, standard responses are sent back to http requests coming from non-interesting users or entities (e.g., researchers investigating this attack from a different source IP address). This also makes the hacks harder to detect and research. The name is derived from predators in the natural world, who wait for an opportunity to attack their prey near watering holes.

Our selected organization (a national CSIRT) did not suffer from such known attack; however, we will test our work into the organization by simulating the same watering hole attack to our selected organization. We will then introduce the same traffic formatted in STIX™and OWL. The reason for this is because this type of attacks is really by-passing the perimeter, being one of the most probable attacks against leading organizations in cybersecurity arena that are usually implementing good solutions at the perimeter. These attacks are targeting the people directly despite the perimeter, attackers try to have some interaction by the end user to help them to get into the system (e.g., like accepting a malicious installer from a trusted source like their usual news Web site)

We will use our integrated and semantic model to manage all related data in an integrated and effective way. Our objective is to better protect our organization risks dynamically by improving our detection and prevention capabilities based on leveraged cyber threat intelligence data.

All relevant information is managed by our model at all levels: data, business logic, services/applications, visualization and risk assessment/risk management.

Another objective it is to facilitate the investigation by a new enriched data model.

For that, all CTI data are parsed into standards STIX™and OWL (stix2.owl). Risk owners will define their services under scope and evaluate their importance as usual (in any other risk management framework like [1]); however, this time they will define that in DRM (drm.owl) not in Excel or any other static file. The rest will be automatically done by our reasoner (e.g., asset dependencies, threats identification, inventory, inherited assessment, etc.)

3.1 TTP of the attack

On July the 8th, a news Web site was compromised. The target organization was used to have traffic to that news Web site from different internal (corporate) users. Since that date (firstSeen=2017-07-08 as timestamp in STIX™), a malicious Javascript artifact (CyberObservable of class artifact and mimetype="javascript" in STIX™) was injected in the homepage, affecting a specific company (e.g., our company) but not any other companies.

That was possible because of the business logic behind the Javascript. All users navigating through http network traffic (NetworkTraffic in STIX™) to that domain name (DomainName in STIX™) where loading the Javascript that was automatically redirecting to load another (second) malicious Javascript. Our company users (and not other connections with a different source IP (srcRef in STIX™) were again

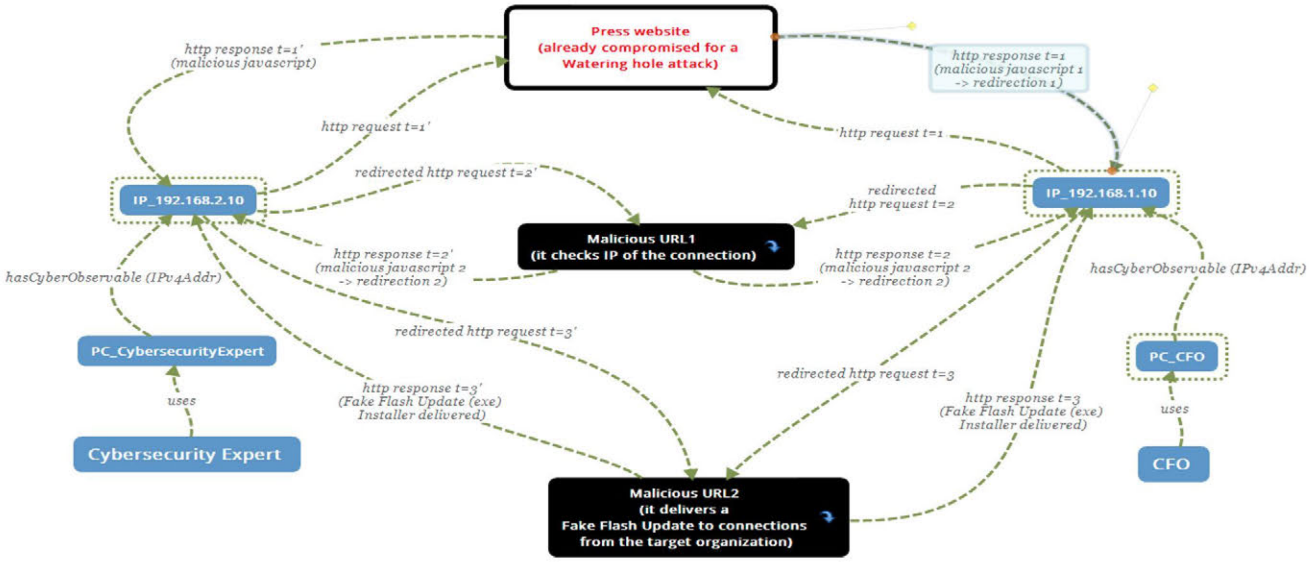


Fig. 14 Graphical description representing the TTP of the watering hole attack

redirected into a different Malicious URL where a Windows (exe) Installer (CyberObservable of Class URL in STIX™) was dropped as a fake Flash update. The rest of the users were redirected to the original Adobe Flash update instead of the malicious one.

The attack started to get victims from its targeted organizations and was persisted at least until August the 1st, where the Javascript was modified again to provide different payloads.

Once a victim executes such a fake update, its personal computer would be compromised, so threat actor would have access to the user data from its personal computer (e.g., stealing credentials, connecting to any other system from its computer, etc.).

A graphical representation of the attack TTP can be seen in Fig. 14.

3.2 The target (and motivation)

The target is to get access to the organization "Classified Data." That means that the attackers are interested in those users with enough access level to any classified data. Attackers could have investigated specific users behavior like the news Web site they are usually reading. This information gathering is usually done by using social networks (e.g., continuous post in Twitter referencing this source). We will simulate the interest of the attackers around some classified data only accessible by specific internal (corporate) users. We will define a data property named hasAccessLevel in our ontology to define the access level of each user of our organization. That property will have a value between 1 and 5 (from low to high-level access, respectively). We can then use this variable in any SWRL rule.

4 Implementation

We used Protegé Tool by Stanford [35] to test our CTI/DRM integrated model with the proposed use case at Sect. 3.

All data related to this TTP description will be under STIX™and OWL. At the same time, all DRM data will be under OWL. As a result, all data in our work are under a semantic data model.

4.1 Architecture

As described in Sect. 2.2, we propose a layered approach with some characteristics:

- The whole architecture is based on a mix of standards (STIX™[4], OWL [10], SWRL [11]) as a standard proposal oriented to have a quick and widespread adoption by the industry with the needed expressivity.
- STIX™ v2.0 standard is used to handle all threat and traffic data (really any CTI data) but under OWL format (stix2.owl is the ontology that we have created for that).
- It is an integrated architecture. Our dynamic risk management framework is leveraging cyber threat intelligence data. For that, DRM.owl ontology imports STIX2.owl. There are relationships between objects and our reasoner together with SWRL will create new knowledge based on those relationships.
- SWRL is used for all business logic: algorithms and rules.
- It is a layered approach, a use case at service / application layer will need some business logic layer to access the data layer. The same happens with DRA/DRM or any other type of service (e.g., an alert).

- It is a semantic approach, which means that a semantic reasoner (in our case, Pellet incremental reasoner [14]) will understand the meaning of data and all their relationships to infer new knowledge dynamically.
- Relevant data (e.g., events, traffic) collected for this work were necessarily parsed into our ontologies to demonstrate the benefits of using semantics. On the other hand, we limited our data collection to STIX™which is becoming a de facto standard. Future sensors will probably collect data under this new standard without the need to parse data (e.g., logs).
- The collection of all data used in a SWRL rule has not necessarily to be done on real time.
- We used a unique central data storage to implement and validate our work.
- The purpose of our work is to demonstrate the benefits to have all data under semantics and standards like STIX™, but it is neither the collection/parsing of the data nor the additional overhead in case of new type of sensors to collect STIX™related data.

Our data model is based on five types of concepts (entities):

- Classes/subclasses to represent the concepts, objects, families or classes of the ontology.
- Object Properties to represent the relationships between objects.
- Data properties to represent the variables, values or fields of each object.
- Data types to represent the types of the data (float, integer, string, dateTimeStamp, custom, etc.).
- Individuals to represent the "instances" of the different Classes.

Because all data will be parsed into semantic OWL data, all the business logic, services/applications, as well as all the dynamic risk assessment and management models are implemented using SWRL rules and a reasoner. We selected Pellet incremental reasoner [14] for this work to provide live semantic reasoning.

As described earlier, we developed STIX2.owl (as the semantic and complete version of STIX™v2.0 [4]) and DRM.owl ontologies. By today, we have about 453 Classes and above 11565 axioms. We also developed different families of SWRL rules to support all levels of the model presented in this work. Thirty-four SWRL rules were created to test all of our implementation.

Visualization model is based in Protegé plug-ins, especially OntoGraph. With regard to querying the data, we used SQWRL as a standard query language by using its corresponding Tab in Protegé [35].

4.2 Instances of our threatened organization

With regard to the organization, we describe the setup to test the behavior against the watering hole attack (use case) in the following subsections. Due to the motivation of the attack (access to classified data), we will describe in detail those instances on risk as well as the corresponding relationships. This will help the reader to understand the attack vector.

Internal (corporate) users: We use two main profiles of the organization, both belonging to class InternalUsers as SubClassOf Users:

> Victim_0 will be the CFO user which is the Chief Financial Officer of the organization. It hasAccessLevel 5 out of 5, so it would have access to all ClassifiedData in our company. However, it will not have so much CybersecurityExperience (3 out of 5), so that characteristic will be taken into account for dynamic risk assessment as this type of users will likely execute a fake SW update with more probability than a cybersecurity expert (like the AdobeFlashUpdate described as part of the TTP).
>
> Victim_1 will be the CybersecurityExpert user which is the Head of cyber threat intelligence of the organization. It hasAccessLevel 3 out of 5, so it would not have access to ClassifiedData in the organization, but it would have access to threat intelligence data and any other type of data equal or below its hasAccessLevel data type. It will have high CybersecurityExperience (5 out of 5), so that characteristic will be taken into account for dynamic risk assessment as this type of users will likely not execute a fake SW update (like the AdobeFlashUpdate described as part of the TTP). Rejecting the execution (rejectExecution action in our model) is semantically equivalent as not executing the fake update. As part of its reaction, it could also create an IOC (indicator of compromise) after detecting and investigating its malicious behavior. It will be shared within the CTI community to avoid future attacks, this time, based on knowns (IOC data). Furthermore, it could also open an internal security investigation together with the CISO (Chief Information Security Officer) and the quality team in charge of the information security certification based on ISO2700X.

Data: It is a subClassOf Asset. We selected five specific instances of data of our organization, all belonging to class data, but only two of them belonging to the subClass ClassifiedData.

There are five instances in our work:

- Data_1 instance has a requiredAccessLevel of 3,
- Data_2 instance has a requiredAccessLevel of 2,

- DataClassified_1 instance has a requiredAccessLevel of 5,
- DataClassified_2 instance has a requiredAccessLevel of 5,
- FileFlashPlayer instance of the malicious executable dropper.

Risk scope is the class representing all the services or projects in scope of the organization risk management certification.

There are different instances of risk scope (RiskScope concept in our ontology) representing all the different services or projects in scope of the risk management certification of the selected organization (based on ISO2700X). By using semantics, our framework will enable interoperability between any risk management framework or standard as all of them will use equivalent concepts.

For the testing of this work, we selected some relevant and interesting services of our organization, especially two services that depends on (property) ClassifiedData (class) named:

- GDPR-compliance instance of class RiskScope, which depends on Data_Cassified_1. This service is
- CyberThreatINTEL-service instance of class RiskScope, which depends on DataClassified_2 and Data_2.

Software is the class representing the installed SW on the PC computers of users to access any data.

To validate this work, we selected the following SW instances:

- OperatingSystem instance of class software representing the operating system installed on the PC,
- Browser instance of class software representing the browser installed on the PC,
- AdobeFlashPlayerPlugin instance of class software representing the browser installed on the PC,

Personal Computer is the class representing the computers of the different users which will be used to access the data.

To validate this work, we selected just the following instances:

- PC-CFO instance of class PersonalComputer representing the PC of the CFO,
- PC-CybersecurityExpert instance of class PersonalComputer representing the PC of the head of cyber threat intelligence.

CyberObservable is the class representing all of the STIX™cyberobservables.

To validate this work, however, we implemented the entire STIX™v2 standard, and our watering hole attack only needed instances belonging to the following subclasses of class CyberObservable:

- Artifacts class representing malicious Javascript artifacts,
- Domain name/subdomains classes representing domains and subdomains,
- IPv4Addr class representing IP addresses,
- URL class representing URLs,
- Network traffic class representing the network traffic.

Safeguards represent the class for countermeasures. They represent the controls setup by the organization as, for example, in equations 1 and 2.

To validate this work, we selected just the following safeguards' instances:

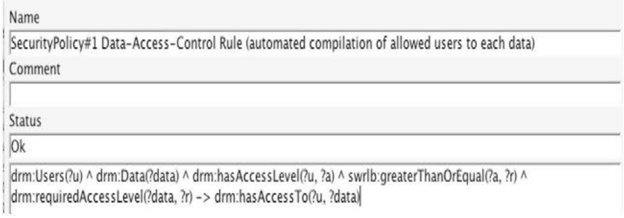
- LDAP-Group-And-Roles-Control instance of class safeguards representing the access control based on LDAP.
- Antivirus instance of class safeguards representing the antivirus of the organization.
- Information sharing control instances which are created automatically by the reasoner for the tactical scenario of Fig. 11.
- HR Related Security Awareness Education and Cyber capabilities control instances which are created automatically by the reasoner for the strategic scenario of Fig. 12.

AssetValuation, Threats, Risk, RiskAssessment, RiskSeverity, SecurityEvents classes are also used; however, their instances are created automatically by using SWRL rules. This is one of our main contributions as errors by hand are frequent in risk management frameworks nowadays. Organizations are also addressing risk management by simplifying relationships because until today they were difficult to consider if done by hand. Our reasoner could understand the meaning of our data, so does its relationships. When risk owners evaluate the service in scope, all related dependencies are considered. An automated dependency tree is created because of static or even inferred relationships. All their values are updated and inherited thanks to automatic asset valuation SWRL rule. The same happen when a datum is of type ClassifiedData, and the reasoner understands that different types of risks are associated with such data because it is classified. Associated risks to that data are identified automatically. Risks are ranging from bad reputation risk (if someone leak that data) to deliberated malicious SW distribution (if someone has access to that data, but it does not have enough cybersecurity expertise to reject a malicious sw distribution intent) depending on the context around. In order to make this happen, rules should allow enough expressivity which is the main reason we use OWL as suggested by STIX white paper (RDF/OWL) [5].

Dependencies are a key factor to define the different scenarios or contexts.

Some examples of SWRL rules have been shown during this work (see Figs. 2, 3, 4, 5, 6, 7). They implement most of the business logic; however, there are different families, and some of them are:

- Asset valuation rules to automatically calculate the asset value based on inheritance (risk frameworks usually evaluate the importance of the service and depending on asset dependency tree, there are cascading effects). This is done automatically to avoid errors (errors by hand are frequent) as well as to have a broader, consistent and a more standardized picture.
- Enrichment rules to enrich all our data model (e.g., for example filling down IP-domain relationships (like a DNS/Reverse DNS). Once some data are missing, relationships between data would help the reasoner to infer the missed data. See Fig. 2 for an example.
- Threat inventory rules to automatically identify threats depending on our topology. As an example, instances of class "Deliberated Malicious SW Distribution Threat" will be created around our company data when specific users that have access to that data have, at the same time, a cybersecurity experience below 3 (3 out of 5). See Fig. 3 to see the SWRL rule.
- Risk inventory rules to make an automatic inventory of all risks. It uses identified threats to guess associated risks to them dynamically. It is important to have enough and accurate information for decision makers along the time.
- Risk assessment rules to make the assessment of each risk which depends on each threat. As an example, see Fig. 16 for residual risk calculation.
- Risk management rules to decide the best strategy depending on the organization policy. An example can be seen in Fig. 10 where we implemented the organization policy for high-level risks in SWRL. In this case, any high-level risk will have associated a mitigation strategy instance. The specific action to be performed could also be automated. Also in this category, we created a tactical level rule (see Fig. 6) to implement a dynamic information sharing once an attack is detected. The whole SWRL detection rule will be shared instead of IOC. On the other hand, a strategic level rule was also created (see Fig. 7) to implement new knowledge for the reasoner. In this case, new awareness training programs are recommended for those users being threatened, and at the same time, they are not experts in cybersecurity.
- Risk severity rules to make another type of automated classification depending on the severity and the policy defined by the organization.



Name

SecurityPolicy#1 Data-Access-Control Rule (automated compilation of allowed users to each data)

Comment

Status

Ok

```
drm:Users(?u) ^ drm:Data(?data) ^ drm:hasAccessLevel(?u, ?a) ^ swrlb:greaterThanOrEqual(?a, ?r) ^
drm:requiredAccessLevel(?data, ?r) -> drm:hasAccessTo(?u, ?data)
```

Fig. 15 SWRL security policy to know data access of users depending on requiredAccessLevel of data and hasLevelAccess of users

- Security policy rules, for example, to automatically make an inventory of users and their data access rights. Depending on hasAccessLevel property of each user and requiredAccessLevel property of each data, the reasoner will keep an updated version of access rights database, that is, who has (or could have) access to what. Thanks to that, the reasoner is able to associate risks in a granular a detailed way. (See Fig. 15)
- Threat intelligence rules to detect security events instances like the SWRL rule shown in Fig. 4. We created it to detect a security event dynamically once it fulfills the antecedent pattern of our rule. The algorithm will detect a malicious TTP pattern when any user from our organization is being redirected after another redirection to an URL which is dropping a (potential malicious) exe installer (in our case a fake Flash installer). By using this behavioral pattern rule, we are detecting a security event related to the TTP of the selected watering hole attack. This security event will trigger a risk re-assessment to recalculate the new risk level automatically. The risk probability will be increased by this security event accordingly and dynamically. As a result, there would be a new instance of class high risk coming from an instance of medium risk due to its probability has changed. The risk is of type deliberated malicious SW distribution. Now it is worse as there has been a security event. An event like this will trigger our reasoner to open or instance a new (proactive) security incident although the end user still has not executed the installer. Once the end user executes it, the proactive incident will become a reactive incident. Recommended actions will differ between proactive or reactive state, but most of them could be automated by our framework.
- Etc.

5 Results

In our case, as the Victim_1 (cybersecurity expert) does not have access to any classified data, its instance will not have access or dependencies from this type of data neither DataClassified_1 nor DataClassified_2. In our model, we can confirm the users having access to classified data anytime by querying the ontology dataset using SQWRL for example. We can check it out also by using any graph visualization plug-in expanding all relationships around any classified data's instances.

The selected watering hole attack was motivated due to the interest of the threat actor to access this type of data, but in our organization only the CFO has access to it. Then, the CFO (Victim_0) is classified automatically by our framework as a potential victim of this type of attack. Different types of risks related to any unauthorized access to classified data will be created by the framework automatically due to the nature of the data (e.g., bad reputation risk when classified data are accessed and leaked, data protection risk, corporate bad image, etc.). In addition to this, once the framework detects that there is an end user which has low cybersecurity experience with access to this classified data, the framework will make automatic connections (relationships) between both types of risks (risk of unauthorized access to classified data and risk of deliberated malicious SW distribution to the user who has access to that data). This type of connections are possible due to different reasons:

– Our model, based on ontologies, SWRL and STIX™, is able to provide enough expressivity to any type of rule that will be understood by the reasoner.
– A SWRL rule is creating automatic dependencies between classified data and the CFO. The rule acts as an access control security policy rule as seen in Fig. 15. The rule understands that if CFO hasAccessLevel of 5 (out of 5), it will have then access to all data, including all classified data.
– A SWRL rule is automatically detecting a potential risk of unauthorized access to classified data due to a potential deliberated malicious SW distribution threat (as an attack vector) associated to the CFO user. This user has high probability to install a malicious SW based on its low experience on cybersecurity. In the selected attack, once the attacker infects the CFO by using a watering hole pattern, there will be an identity theft granting access to any classified data. This complex TTP pattern is now possible to be written as a threat detection algorithm by using our semantic framework. (See the SWRL rule in Fig. 3 as an example)
– Although the CFO becomes our main target as victim due to the TTP of this attack, other relevant staff with similar access level from the organization could also be a potential target of the ThreatActor. Then, our GDPR-compliance service in scope of our DRA/DRM will identify this dependency as well. On the other hand, the CyberThreatINTEL service is also dependent of classified data as well as non-classified data. This service will have the same type of risk associated with it. In this case, the reasoner will explain that the risk comes from the probability to get access to Data_Classified_2 in case the CFO is being hacked by a deliberated malicious SW distribution threat.
– Based on the same security events, our reasoner initiates the sharing of the SWRL rule to detect such TTP pattern within other offices and partners. It is a shift of paradigm because IoCs are not shared (they are simple to be changed by attacker) but the intelligence algorithm itself. Knowledge about how to detect specific patterns is shared. This is an implementation of a tactical level risk management action. See SWRL rule at Fig. 6 and created instances at Fig. 11.
– Also because of the same detection, our reasoner proposes other action at strategic level. In this case, specific awareness training for workforce capacity building is selecting only the users being threatened which at the same time has low cybersecurity experience. See SWRL rule at Fig. 7 and created instances at Fig. 12.

DRA/DRM is calculated properly as expected and risks are classified into different risk management strategies depending on the severity.

Victim_0 and Victim_1 browsed the Web site the day before it was compromised, but no security alerts were received (instanced). The rules were created, but that malicious behavior was inexistent.

The day after, when the press site was compromised, both users, as usual, started navigating to the infected domain, but after the homepage was loaded, an injected Javascript started to load different artifacts from different URLs making different redirections. After that, a payload was dropping a fake (exe) AdobeFlashPlayer installer as expected in our work to both users.

Once our SWRL rule detected that traffic, it created a new instance of type SecurityEvent class classifying that behavior into a security event of type "Dropper behavior of Malicious Windows Executable," again for both users. The rule is then a pattern-like to detect the TTP of our watering hole attack.

The SWRL TTP-like rule was designed by using a combination of sequential http request and response traffic analysis while being automatically redirected by malicious Javascript until an URL is dropping a fake windows installer.

All security events triggered were using our proposed semantic architecture. Then, it was very easy to follow all new or established relationships. As an example, knowing the LAN IPv4Addr (CyberObservable) originating the http request connection, we could not only identify the user behind that connection but we were also able to identify all related risks (and only those related to this specific threat) to make a dynamic re-assessment (DRA). Risks related to other type of threats were kept unchanged as they did not have relationships within these type of security events.

In our work, however, we had security events triggered from both users' network traffic (both were navigating and experiencing the same attack), our system considered that the Cybersecurity Expert was likely not going to execute the strange (fake) Flash update or any other strange update due to its cybersecurity experience. Risk re-assessment only took place for those risks which were belonging to those users whose experience in cybersecurity were less than 3 (out of 5). The model was then able to accommodate such kind of specificities. More than that, if our CFO improves its cybersecurity experience in the coming future, once the value is updated in its end user instance, our framework will adapt itself to the new context. Some risks will then be removed as the same SWRL rules will consider that some past risks are no longer justified. The reasoner will identify inconsistencies over the data anytime.

We are then moving forward from the original (static) risk assessment approach used by the organization until today (see Eq. 1) to a more feasible, meaningful, more effective, realistic, complete and dynamic approach (by implementing Eq. 2).

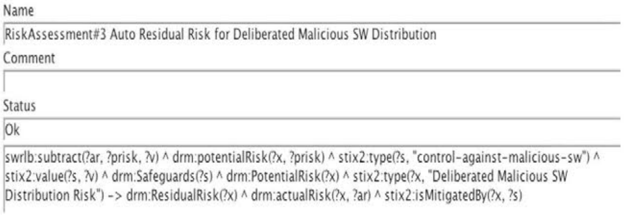
For this evaluation, we limited the implementation mostly to "Deliberated Malicious SW Distribution Risks" (Fig. 16) but also to other type of non-IT-related risks, like strategic risks of type "Bad Reputation Risks" because their relationships in case of potential data leak when unauthorized access to classified data take place.

Again, due to this semantic approach, it is easy to follow risks relationships from a service/project to the users and vice versa, even to connect its network traffic to a specific service in risk dynamically.

In our example, IP 192.168.1.10 belongs to our CFO user. In case it executes the fake Flash update, a new rule would be able, for example, to create an automated incident response with all the related context (security event, user potentially infected, dateTimeStamp, Services Affected, etc.). Without a malware analysis yet, or without more cyber threat intelligence data, the incident could not have still the highest severity score (we still do not know if the dropped file is a malware and their motivation); however, it is a malicious pattern detected.

But, when the binary file is identified as malware, the incident severity would then get the highest score as classified data could have been compromised already by using stolen credentials.

One of the main challenges in cybersecurity it is to work against unknown or emerging threats (e.g., APT) having real-time visibility about our risk exposure along the time. Another challenge is to clearly define when a security event becomes an incident (preventive or reactive).



Name

RiskAssessment#3 Auto Residual Risk for Deliberated Malicious SW Distribution

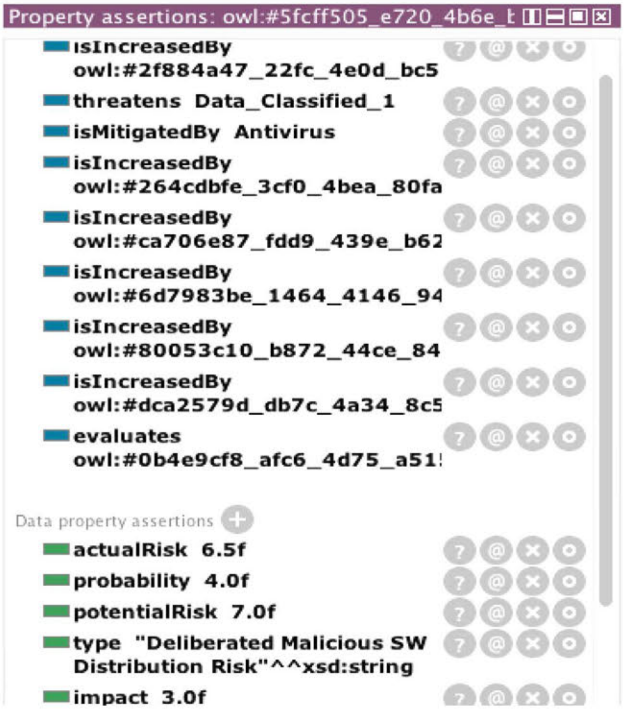
Comment

Status

Ok

```
swrlb:subtract(?ar, ?prisk, ?v) ^ drm:potentialRisk(?x, ?prisk) ^ stix2:type(?s, "control-against-malicious-sw") ^
stix2:value(?s, ?v) ^ drm:Safeguards(?s) ^ drm:PotentialRisk(?x) ^ stix2:type(?x, "Deliberated Malicious SW
Distribution Risk") -> drm:ResidualRisk(?x) ^ drm:actualRisk(?x, ?ar) ^ stix2:isMitigatedBy(?x, ?s)
```

Fig. 16 SWRL residual risk calculation for deliberated malicious SW distribution risks when safeguards of type "control-against-malicious-sw" are available. Safeguards reduce potential risk



Property assertions: owl:#5fcff505_e720_4b6e_t

isIncreasedBy owl:#2f884a47_22fc_4e0d_bc5

threatens Data_Classified_1

isMitigatedBy Antivirus

isIncreasedBy owl:#264cdbfe_3cf0_4bea_80fa

isIncreasedBy owl:#ca706e87_fdd9_439e_b62

isIncreasedBy owl:#6d7983be_1464_4146_94

isIncreasedBy owl:#80053c10_b872_44ce_84

isIncreasedBy owl:#dca2579d_db7c_4a34_8c5

evaluates owl:#0b4e9cf8_afc6_4d75_a51!

Data property assertions

actualRisk 6.5f

probability 4.0f

potentialRisk 7.0f

type "Deliberated Malicious SW Distribution Risk"^^xsd:string

impact 3.0f

Fig. 17 Screenshot of Protegé tool [35] of an instance of class risk assessment and type "Deliberated Malicious SW Distribution Risk" whose risk has being increased automatically by different security events as well as mitigated by one countermeasure, the antivirus

By using our framework (as shown in Fig. 17), we have all the needed expressiveness to better know what is really happening along the time; in this case, we know that there is a risk automatically identified of type deliberated malicious SW distribution which has been mitigated by one safeguard, but at the same time, it was increased by different security events. We perfectly know the connection of this risk to the affected assets and services, and all information is consistent. We can query our model to know more about all the relationships and reasoner conclusions, but we can also use interactive graphs to see all the relationships as shown in Fig. 1. Apart from operational level, our framework is able to work at tactical and strategic levels as seen in Figs. 11, 12.

6 Conclusion

Today, risk assessment (RA) and management (RM) are mostly manual processes performed once per year by different experts based on their personal opinions. On the other hand, any entity is exposed to cybersecurity threats everyday. Unfortunately, these threats are not taken into consideration dynamically into the organization risk calculation. Risk exposure level calculation, countermeasure's responses, projects or related investment plans should be updated dynamically and proportionally to the threat level and risk exposure of each organization along the time. There should not be treated as a static annual review process (by auditors).

We developed a formal model based on standards to connect real-time threats to risk calculation and risk management processes which also provide better automation, enrichment, detection capabilities and simplicity by using standards STIX™ [4], OWL [10], SWRL [11] and a reasoner [14].

This paper presents the first practical DRA/DRM approach applied to up-to-date threat and risk processes of an international reference entity, a national CSIRT. We have selected a real publicly known attack for the implementation, as we consider it a good example to test our proposal on leading organizations that could easily be impacted by this type of attack due to its nature. This type of attacks is very difficult to detect even by leading organizations. We implemented behavioral pattern rules in SWRL to detect and update our risk level exposure accordingly with so much expressivity to understand what is really happening either by humans or machines. At the same time, we demonstrated how a specific security event could trigger different actions beyond the operational level, like the tactical and strategic levels. In our case, at tactical level, the same attack produces an automatic risk intelligence sharing (share of TTP detection algorithm but not specific IoC) as a tactic to avoid bigger impact of a potential campaign against other remote offices or partners even if the IoCs are changed during the attack. As a future research direction, there is a need to improve incentives for intelligence sharing (IoC or algorithms). At strategic level, specific awareness training sessions were identified to those victims involved in the attack which at the same time have poor cybersecurity knowledge.

Compliance with ethical standards

Conflict of interest All authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. ISO/IEC 27005:2008, Information technology—security techniques and Information security risk management (2008)
2. ISO 31000:2018, Risk management—guidelines (2018)
3. Bianco, D.: "The Pyramid of Pain". http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html (2014). Accessed 15 July 2018
4. OASIS: "STIX™ 2.0 specifications". https://oasisopen.github.io/cti-documentation/resources#stix-20-specification. Accessed 7 Aug 2018
5. OASIS: "STIX™ White paper". https://stixproject.github.io/about/STIX_Whitepaper_v1.1.pdf. Accessed 15 June 2018
6. OASIS: "TTP (Techniques, Tactics and Procedures" by STIX™. https://stixproject.github.io/getting-started/whitepaper/#tactics-techniques-and-procedures-ttp. Accessed 7 Aug 2018
7. OASIS: "Campaigns by STIX™". https://stixproject.github.io/getting-started/whitepaper/#campaigns. Accessed 7 Aug 2018
8. OASIS: "Incidents by STIX™". https://stixproject.github.io/getting-started/whitepaper/#incidents. Accessed 7 Aug 2018
9. European Commission and European Parliament: "NIS Directive". http://data.europa.eu/eli/dir/2016/1148/oj. Accessed 7 Aug 2018
10. W3C: "OWL". https://www.w3.org/OWL/. Accessed 1 June 2017
11. W3C: "SWRL Semantic Web Rule Language". https://www.w3.org/Submission/SWRL/. Accessed 1 June 2017
12. W3C: "Ontology". https://www.w3.org/standards/semanticweb/ontology. Accessed 1 June 2017
13. W3C: "Inference". https://www.w3.org/standards/semanticweb/inference. Accessed 1 June 2017
14. W3C: "Reasoner". https://www.w3.org/2001/sw/wiki/Category:Reasoner. Accessed 1 June 2017
15. W3C: "Pellet reasoner". https://www.w3.org/2001/sw/wiki/Pellet. Accessed 1 June 2017
16. Herzog, A., Shahmehri, N., Duma, C.: An ontology for information security. Int. J. Inf. Secur. Priv. **1**(4), 1–23 (2007)
17. Ekelhart, A., Fenz, S., Klemen, M., Weippl, E.: Security ontologies: improving quantitative risk analysis. In: Proceedings of the 40th Hawaii International Conference on System Sciences (2007)
18. Fenz, S.: Ontology-based generation of IT-security metrics. In: Proceedings of the 41st Hawaii International Conference on System Sciences (2008)
19. Goluch, G., Ekelhart, A., Fenz, S., Jakoubi, S., Tjoa, S., and T. M.: Integration of an ontological information security concept in risk-aware business process management. In: Proceedings of the 41st Hawaii International Conference on System Sciences (2008)
20. de Vergara, J.E.L., et al.: A semantic web approach to share alerts among security information management systems. Commun. Comput. Inf. Sci. **72**, 14–25 (2010)
21. Mateos, V., Villagrá, V.A., Romero, F.: Ontologies-based automated intrusion response system. Comput. Intell. Secur. Inf. Syst. **2010**, 99–106 (2010)
22. Obrst, L. et al.: MITRE—developing an ontology of the cyber security domain. In: MITRE (2012)
23. Singapogu, S. et al.: Security ontologies for modeling enterprise level risk assessment. In: 2012 Annual Computer Security Applications Conference, Orlando (2012)
24. Erbacher, R.F.: Ontology-based adaptive systems of cyber defense. In: Semantic Technology for Intelligence, Defense and Security Conference, Fairfax, VA (2015)
25. Syed, Z. et al.: UCO—unified cybersecurity ontology. In: The Workshops of the Thirtieth AAAI Conference on Artificial Intelligence. Artificial Intelligence for Cyber Security: Technical Report WS-16-03 (2016)

26. Gao, P. et al.: AIQL: enabling efficient attack investigation from system monitoring data. In: USENIX Annual Technical Conference (2018)
27. Gao, P. et al.: SAQL: a stream-based query system for real-time abnormal system behavior detection. In: USENIX Security Symposium (2018)
28. Meszaros, J., Buchalcevova, A.: Introducing OSSF: a framework for online service cybersecurity risk management. Comput. Secur. **65**, 300–313 (2017)
29. Qamar, S., Anwar, Z., Ashiqur Rahman, M., Al-Shaer, E., Chu, B.-T.: Data-driven analytics for cyber-threat intelligence and information sharing. Comput. Secur. **67**, 35–58 (2017)
30. Poolsappasit, N., Dewri, R., Ray, I.: Dynamic security risk management using Bayesian attack graphs. IEEE Trans. Dependable Secure Comput. **9**(1), 61–74 (2012)
31. Schiffman, M.: Common vulnerability scoring system (CVSS). http://www.first.org/cvss/cvss-guide. html (2011)
32. Mozzaquatro, B.A. et al.: An Ontology-Based Cybersecurity Framework for the Internet of Things, Sensors (Basel, Switzerland), vol. 18, 9 3053 (2018)
33. Zhang, J., Yang, J., Li, J.: When rule engine meets big data: design and implementation of a distributed rule engine using spark. In: IEEE Third International Conference on Big Data Computing Service and Applications. BigDataService), San Francisco, CA (2017)
34. Alrwais, S., Yuan, K., Alowaisheq, E., Liao, X., Oprea, A., Wang, X., Li, Z.: Catching predators at watering holes: finding and understanding strategically compromised websites. In: Proceedings of the 32nd Annual Conference on Computer Security Applications (2016)
35. Stanford University “Protege”. https://protege.stanford.edu/