NUST Journal of Engineering Sciences, Vol 9, No 2, 2016, pp 42-49

Towards a Generic Model for Risk Analysis of the Internet of Things (IoT)

Mujahid Mohsin[1], Zahid Anwar[1,2] and Farhat Zaman[1]
[1]National University of Sciences and Technology (NUST), Islamabad, Pakistan
[2]Fontbonne University, St. Louis, Missouri, USA
e-mail: {13phdccsmmohsin, zahid.anwar, 13msccsfzaman}@seecs.edu.pk, zanwar@fontbonne.edu

Received: 07 September 2016 Accepted: 08 November 2016

Abstract

The Internet of Things (IoT) has spurred the interaction of a multitude of smart physical objects with the existing cyber world. These connected "things" leverage heterogeneous protocols, diverse capabilities and complex environmental interdependencies, which have reshaped their risk profiles through introduction of novel threat vectors. In this paper, we present a formal framework to model and analyze the security risks linked with generic IoT systems. The approach uses existing and widely-accepted Web Ontology Language (OWL) based ontologies, by extending them with IoT-specific concepts and populating them with IoT instances. Risk assessment, quantification and selection of viable mitigation techniques is carried out automatically with the help of rule-based constraints and queries applied over OWL knowledgebase. The practicality and effectiveness of the approach is verified through implementation and evaluation over realistic IoT systems.

Keywords: Internet of Things, Automated risk analysis, IoT security modeling, IoT ontology, OWL, SWRL.

Introduction

The Internet of Things (IoT) has given us a notion of a smartly connected world being driven by autonomous network of "things"; observing, interacting and implementing features with minimal human intervention. The concept has received a wide-acceptance, giving birth to promising applications in all technological domains, ranging from common home and personal appliances to sophisticated systems such as linked with e-Health, industrial automation and other safety-critical infrastructures. The number of connected devices has already surpassed human population and is predicted to reach 50-100 billion by the year 2020 [1]. The goal of the IoT is to "*enable things to be connected anytime, anyplace, with anything and anyone ideally using any path/network and any service*" [2]. Such a flexible communication model, coupled with the ever-expanding size, diversity and sophistication of IoT devices, introduces several new threat vectors, thus significantly changing the security risk profiles. Security is considered only as strong as the weakest link in the system. Therefore, protection of the IoT infrastructures requires a thorough risk analysis of interlinked IoT devices for ensuring that they are properly-configured, standard-compliant and offer adequate resiliency against traditional as well as IoT-specific threats. Manual approaches for risk assessment and management for large scale IoT systems are not only taxing, they are also prone to ill-judgements and human-errors. The situation calls for smart risk analysis and security planning solutions, which can be driven by automation and machine intelligence through understanding of risk semantics.

Realizing the emerging security challenges and the proposed solution as discussed above, in this paper, we present a formal ontology-based approach for automated risk analysis of complex IoT systems. Our research methodology can be broadly decomposed into two main steps. Initially, the security capabilities and operational dependencies of individual IoT devices are semantically registered in a knowledgebase, defined as a conjunction of two ontologies i.e. IoT ontology and security ontology. We have used Web Ontology Language (OWL) [3] to encode the semantics of the domain knowledge. After the semantic registration, security soundness and risk exposure of registered facts is derived with the help of rule-based constraints and queries, developed using the built-in features of Semantic Web Rule Language (SWRL) [4] and Semantic Query-enhanced Web Rule Language (SQWRL) [5], respectively. The inferred knowledge is also used to isolate high-risk devices and short-list viable security solutions, which can be used to mitigate the identified risks. The developed framework can therefore, be employed both at IoT design and integration stages and can assist to gauge as well as restraint the system-level security risks.

The proposed approach renders the following key benefits: (a) Ontology based knowledge is hierarchical and hence reusable. We have developed our work by extending existing ontologies and our work can also be extended and reused in other associated domains of IoT. (b) OWL being highly expressive, allows defining system knowledge with high complexities and constraints as compared to other approaches such as object oriented methods, database management systems and constraint satisfaction approaches (c) Knowledge semantics defined in OWL can be automatically reasoned to remove inconsistencies and infer new knowledge, consistent with the global model (d) OWL and SWRL based system models are independent of actual implementation. Therefore, our approach is flexible in choosing implementation platforms (such as Jena, Jess, Prolog, etc), without the need to change the core model.

The rest of the paper is organized as follows. In Section 2, we present a review of related work. The contributions in the domain of ontology extension and alignment are presented in Section 3. Section 4 discusses different security constraints in the form of selected rules and queries.

Lastly, Section 5 presents the implementation and evaluation, while Section 6 concludes the paper, with pointers to the on-going future work directions.

Literature Review

Our work benefits from related research in ontologies for the IoT and security capabilities as well as the use of risk management approaches for IoT security. This section summarizes the state-of-the-art in relevant areas.

Use of Ontologies for the IoT

Application of ontologies in the domain of IoT is an emerging research area. The existing literature mainly targets the efficiency, scalability and interoperability aspects of IoT, with limited focus on its security issues.

Sensor networks claim a major share of typical IoT systems. A few noteworthy ontologies have been proposed to model various aspects of connected sensors. Compton et al. [6] presented an OWL ontology for reasoning and querying about sensors and observations. Calder et al. [7] used ontologies about sensor-packages and constraints defined as rules to reason on real-time sensor data and detect data anomalies and unexpected conditions. SSN ontology [8], developed by W3C Semantic Sensor Network Incubator Group (W3C-XG), is one of the recent and more formal efforts to comprehensively model sensors and their observations. Being generic and domain-independent, the SSN ontology integrates most of the concepts of earlier ontologies and models sensors' capabilities, deployment, operating restrictions and the measurement process. However, none of the contributions mentioned above addressed the security related aspects of sensor networks.

Modeling IoT Entities: Typical IoT systems constitute a diverse set of entities such as sensors, actuators, appliances, fixed and mobile controllers and tag devices. Only a few efforts can be found in literature, which capture the knowledge of IoT entities and their interactions. IoT-Lite [9], a lightweight ontology, represented IoT objects, resources and services. This ontology is primarily focused on sensing, though it introduces some higher level concepts of actuation as well. De et al. [10] proposed a suite of three ontologies modeling entities, resources and services in the IoT domain. IoT-O ontology [11] presented an integration of multiple ontology modules covering the sensing, actuation, life-cycle, energy and service aspects of IoT. While their sensing module leveraged the SSN ontology, the actuation module was developed separately to cover the behavioral patterns of IoT actuators. However, none of the ontologies discussed above modeled the security vulnerabilities and risk profiles of IoT systems as a whole. Moreover, they are also limited in capturing the holistic behavior of IoT device-device and device-environment couplings.

Ontology-Driven IoT Security: Gyrard et al. [12] designed a new ontology-based security knowledge termed as Security Toolbox:Attack and Countermeasures (STAC) for satisfying the security requirements of ETSI Machine to Machine (M2M) architecture. STAC categorizes security mechanisms and attacks based on IoT communication mediums. The main goal of STAC is to target individual IoT devices (rather than integrated systems), for motivating the designers to embed security during the design process. Moreover, the published ontology version of STAC does not cover the IoT-specific instances of security mechanisms and protocols, thus limiting its reusability profile.

IoT-Specific Risk Analysis

With the rise in IoT-specific security breach incidents, the field of risk assessment and management for IoT related threats has also emerged as a dedicated research area. Liu et al. [13] proposed a dynamic risk assessment methodology for the IoT, inspired by the artificial immune system. Their approach computed the changing risk value of an IoT system based on attack intensity, as measured by different attack detection agents. In another work [14], researchers discussed the security risks being contributed by the ever-increasing influx of IoT devices. The authors critically analyzed such emerging risks, their root causes and viable mitigation techniques. Questionnaire-driven empirical study is another way of quantifying the security risks. Chang et al. [15] utilized this approach to investigate enterprise risk factors for governing the risk of IoT environments. Jacobsson et al. [16] conducted an empirical and scenario-based risk analysis for smart home automation systems. Such empirical analyses are mostly manual and their findings are based on experiences and views of the experts. Our ontology-based approach can leverage and extend such manual methods to automatically reason about risk applicability and countermeasures, not only on individual IoT entities but can also deal with complex and large-scale IoT systems.

Modeling IoT and Security Concepts

In this section, we discuss the contributions made during the ontology engineering phase by presenting the salient features of developed ontologies. The section begins with introduction to our ontology engineering approach, followed by a discussion on important ontology concepts.

Ontology Engineering Approach

Our research approach is focused on reuse of existing ontologies, wherever possible by extending them with IoT-specific features such as abilities to sense, control, identify and impact applicable features, pertaining to both cyber and physical worlds. After a careful review of available options for IoT ontologies, we shortlisted the SSN ontology [8] as the most suitable candidate to serve as our foundation stone. The SSN ontology models operational aspects of connected sensors and adopts a modular approach to offer reuse of ontology from different perspectives, including our desired views of data and observation, property and features of interest as well as broader system's perspectives.

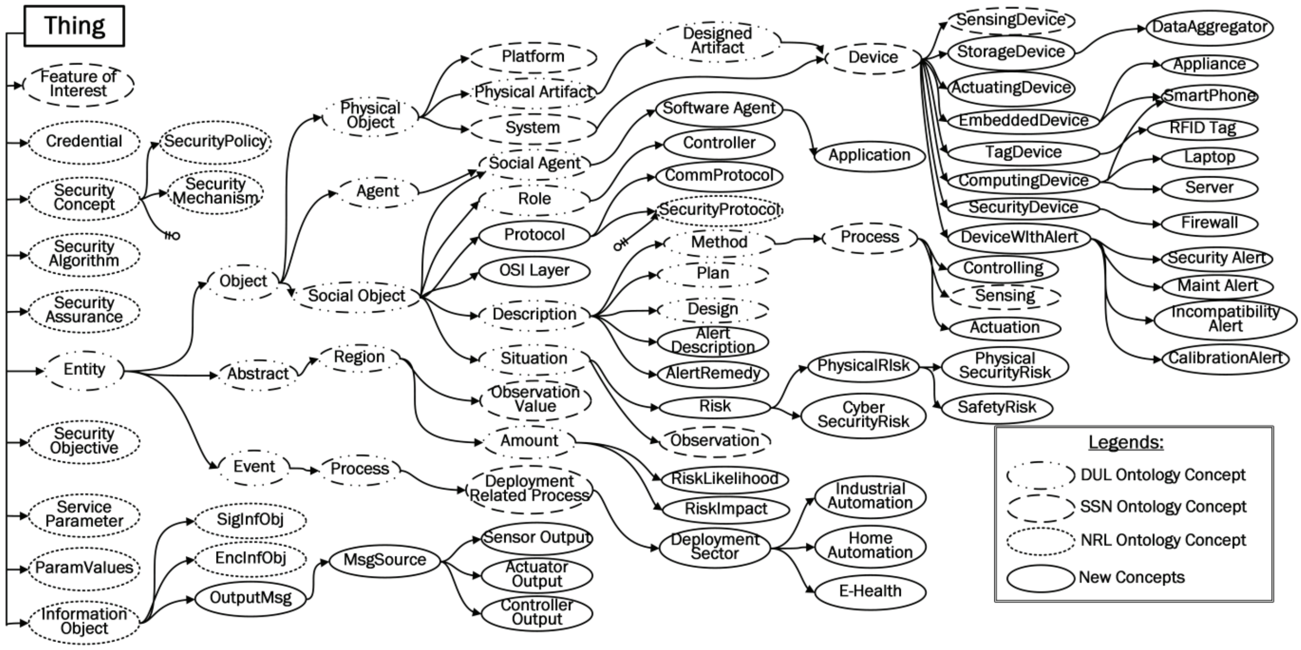

Fig. 1: Sub-Class View of the Aligned IoT and Security Ontologies

Additionally, SSN is already aligned with the relevant concepts of Dolce Ultra-Light (DUL) ontology [17], a light weight ontology to describe generic concepts across multiple domains. This further enhances coherence, consistency and reusability of SSN structure. However, SSN does not cover semantic definitions of other IoT devices such as actuators, controllers and identity devices. Moreover, it does not aim to model any security properties of sensors. Therefore, to support our work, we extended the SSN ontology in two major directions: (a) Transforming SSN into an IoT ontology with an aim to formally capture the interactions and dependencies among different IoT and environmental entities. (b) Alignment of SSN with a suitable security ontology to reason about the security capabilities and requirements for analyzing IoT-specific risks.

With regards to annotation and reasoning over security properties, we have extended the NRL security ontology [18]. The reason to select NRL is its flexibility and interface simplicity. NRL is a collection of seven security ontologies each covering a unique security domain such as security credentials, security algorithms, security assurance and security concepts (includes security protocols, mechanisms and policies) as well as dedicated ontologies for querying and linking with OWL-S service ontology [19]. The ontologies are well-integrated through use of inter-ontology properties and also mention a list of security objectives which are met by the linked security concepts. However, NRL being generic in nature does not define IoT-specific security requirements and capabilities. Moreover, it does not model security risks, entity-specific vulnerabilities and threats. Therefore, we extended the NRL ontology by adding these missing features.

The resulting IoT (extended SSN) and Security (extended NRL) ontologies were mutually aligned using the inter-ontology relationships. Additionally, similar to SSN, we have used Dolce Ultra-Light (DUL) [18] as a parent ontology to explicitly define and align the novel concepts using a common foundation. The overall subclass view, depicting important concepts of the integrated ontologies, is given in Fig. 1. Since DUL, SSN and NRL ontologies are very extensive and have been comprehensively discussed in the referred literature; therefore, re-introducing their structure is beyond the scope of this paper. Hence, the figure draws only the relevant concepts of DUL, SSN and NRL ontologies alongside the newly added concepts and differentiates them with appropriate legends. The new concepts are prefixed as "iot:" to distinguish them from existing concepts. The ontological structure and design considerations of some of the important concepts and properties are discussed in the subsequent sub-sections. Few additional concepts and properties not discussed in this section are introduced in Section 4, while explaining the security constraints.

IoT Ontology

Fig. 2 represents the relational view of new IoT and security concepts with existing concepts. It links these concepts with the help of appropriate OWL properties. Both Figures 1 and 2 are to be consulted in coherence to grasp the overall structure of the new ontologies. Some important extensions made in the SSN ontology are introduced below:

IoT Devices

Different categories of IoT devices are added as sub classes of *ssn:Device* alongside *ssn:'Sensing Device'* as shown in Fig. 2. Hence, these devices inherit the properties of *ssn:System* and *dul:'Physical object'*.

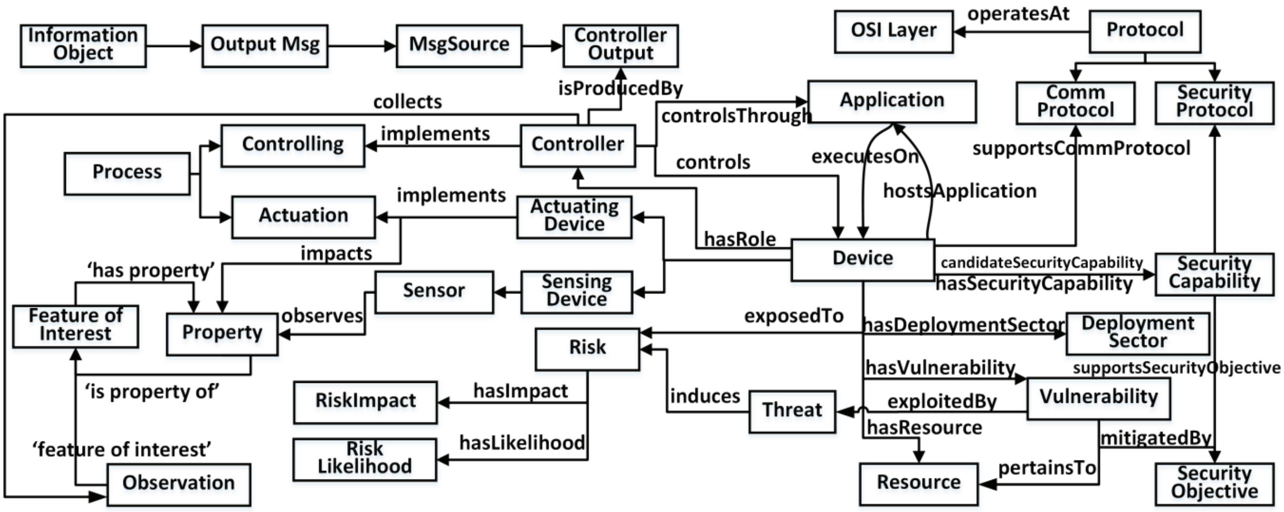

Fig. 2: Relational View of the Resulting Ontology (Arrows not labelled represent sub-class relationships).

Besides inheriting the properties from their super-classes, each device type is also defined by its unique set of properties. To illustrate this further, semantic definitions of *iot:ActuatingDevice* and *iot:SecurityDevice* are given below:

ActuatingDevice: Actuators are known for their capabilities to impact the associated environmental properties of respective features of interest such as the light bulb changing the luminosity (property) of the living room (feature of interest) or the air-conditioner impacting on the room temperature. Thus, the *iot:ActuatingDevice* class is linked with *ssn:Property* through *iot:impacts* object property. As a subclass of *ssn:Device* and *dul:'Physical object'*, *iot:ActuatingDevice* inherits their properties as well. Therefore, actuators can be characterized by their respective operating and survival ranges and they may also constitute other devices as sub-systems (through *ssn:'has subsystem'* property).

SecurityDevice: The *SecurityDevice* class is defined as a subclass of *ssn:Device* and an equivalent class to $\exists\, ssn{:}implements \cdot nrl{:}SecurityMechanism$ (i.e. any device implementing some security mechanism is categorized as a security device and all security devices implement some security mechanism). Different security devices such as *Firewall, IDS* and *IPSec* are added as sub-classes of *SecurityDevice*. Each such subclass can be further extended to align corresponding ontologies of the respective security devices, depending on the requirements.

Controller

We define controllers as those IoT devices which can be used to (a) configure other devices directly or through authorized applications or (b) force other devices to perform certain actions based on the commands generated by them. Controllers play a crucial role in any IoT network as they are primarily used to exercise administrative privileges over devices being controlled. The C*ontroller* class is defined as a sub-class of *dul:Role*. This role can be assigned to any appropriate device through *iot:hasRole* property. Controllers *iot:collects* the observations made by linked sensors and based on the collected information, control authorized IoT devices. Therefore, the concept is linked with *ssn:Device* through *iot:controls* property. They normally use some *iot:Application* for *Controlling*; therefore, *Controller* and *Application* are defined as domain and range, respectively, for property *iot:controlsThrough*.

Deployment Sector

DeploymentSector is introduced as a sub class of *ssn:'deployment related process'* which links the instances of systems or devices with instances of sectors (e-health, home automation, industrial automation, smart vehicles, etc.) in which they are deployed. This discrimination is important since the same device, deployed in different sectors, presents varying level of security risks. For example, consider the risks attached with a temperature sensor installed for home automation and the same sensor installed at a nuclear plant. The types, impacts and motives behind the attacks on these two devices may widely differ. The concept *ssn:System* is linked with *iot:DeploymentSector* through property *iot:hasDeploymentSector*.

Security Ontology

The IoT ontology discussed above is required to be aligned with some suitable security ontology to reason about the security capabilities and risk profiles of IoT systems. As justified earlier, we have used the NRL ontology for this purpose after enriching it with IoT-specific security requirements and associated concepts. Some of the worth-mentioning enhancements made to the ontology are discussed here.

Vulnerability and Threat

The NRL ontology does not cover semantic definitions of security vulnerabilities and threats. However, risk analysis of IoT systems essentially require these concepts to be properly defined and precisely aligned with the existing security concepts. Therefore, we define vulnerability and threat as dedicated classes inside the security main ontology of NRL ontology suite. Fig. 2 illustrates the key ontology relationships for these concepts. *Vulnerability* class points to specific weaknesses present in a given IoT device. Each vulnerability *pertainsTo* software, firmware, hardware, communication protocol/link or information (data); categories defined as subclasses of the *iot:Resource* class. These vulnerabilities can be mitigated by implementing corresponding security objective(s). For example, the vulnerability of inadequate confidentiality and authentication at an IoT gateway can be countered by implementing the security objective of *end-to-end-security* among the communicating nodes, using that gateway. Thus, vulnerabilities in a system can be mapped to the corresponding security capabilities, which can counter them through a common security objective. Vulnerabilities are *exploitedBy* security threats such as a lack of authentication at actuator level can be exploited to manipulate or fabricate the commands directed towards it.

Security Risk

We assess security risk as a function of threats and vulnerabilities contained by all the devices constituting an IoT system. Since the nature of risks keeps on evolving over time, owing to the dynamic behavior of threats, we define *Risk* as a subclass of *dul:Situation*. In addition to routine *CyberSecurityRisk*, IoT infrastructure is also susceptible to *PhysicalRisk*. Therefore, both these risk categories are introduced as sub-classes of *Risk* (Fig. 1). IoT related risks can be quantified and analyzed with reference to *RiskLikelihood* and *RiskImpact*, both defined as sub-classes of *dul:Amount*. Instance population of various risk categories and their respective scores can leverage domain specific risk studies. As a proof of concept, we have adopted the risk analysis conducted by Jacobsson et al. [16] for smart home automation systems. The authors in this study, identified a total of 32 risks and categorized them with reference to associated threats, vulnerabilities and severity, based on the likelihood and impact of the exploit.

IoT Security Objectives

The existing list of *nrl:SecurityObjectives* defined in the Security Main ontology is extended with following IoT-specific and self-explanatory security objectives a) *secure-firmware-upgrade* b) *secure-bootstrap* c) *end-to-end-security* d) *group-key-management* e) *host-mobility* f) *device-authentication*

Adding New Instances

Existing version of the NRL ontology is not populated with instances of IoT-specific security protocols. We conducted an extensive literature review of such security protocols and added some of the most widely-adopted and recommended protocols as instances of the appropriate NRL concepts. Moreover, existing NRL instances were also enriched by annotating them with data linked through newly added properties.

Rule-Based Reasoning

This section gives an overview of the proposed methodology for ontology-driven risk assessment and management of IoT systems. The dependencies, risks and corresponding security requirements are derived from the existing OWL facts by using suitable rule and query languages (such as SWRL and SQWRL, respectively). Rule-based reasoning leverages existing or built-in OWL concepts and properties to offer more powerful deductive reasoning capabilities than OWL alone. We initially deploy SWRL to infer device level pairings based on their dependencies and matching security protocols and categorize these rules as inherent constraints. Followed by that, we demonstrate sample rules to infer risk exposure, quantify its severity and propose viable mitigation options.

Inherent Constraints

These are the constraints, which derive dependency and security relationships among the registered IoT entities.

Rule-3: Mapping Vulnerabilities
$ssn{:}\,Device(?\,d1) \wedge ssn{:}\,Device(?\,d2)$
$\wedge\ iot{:}\,hasDependency(?\,d2, ?\,d1)$
$\wedge\ iot{:}\,noEncrptionPairing(?\,d2, ?\,d1)$
$\Rightarrow\ hasVulnerability(?\,d1, poor\_confidentiality) \wedge$
$hasVulnerability(?\,d2, poor\_confidentiality)$
Rule-4: Risk Exposure
$ssn{:}\,Device(?\,d) \wedge iot{:}\,hasVulnerability(?\,d, v) \wedge$
$iot{:}\,exploitedBy(?\,v, ?\,t) \wedge iot{:}\,Threat(?\,t) \wedge$
$iot{:}\,induces(?\,t, ?\,r) \wedge iot{:}\,Risk(?\,r)$
$\Rightarrow\ iot{:}\,exposedTo(?\,d, ?\,r)$
Rule-5: Risk Scoring
$ssn{:}\,Device(?\,d) \wedge iot{:}\,exposedTo(?\,d, ?\,r) \wedge iot{:}\,ha\_$
$sLikelihood(?\,r, ?\,l) \wedge dul{:}\,hasDataValue(?\,l, ?\,lv)$
$\wedge\ iot{:}\,hasImpact(?\,r, ?\,i) \wedge dul{:}\,hasDataValue$
$(?\,i, ?\,iv) \wedge swrlb{:}\,multiply(?\,R, ?\,lv, ?\,iv) \wedge$
$swrlb{:}\,greaterThanOrEqual(?\,R, 10.0)$
$\Rightarrow\ iot{:}\,CriticalDevice(?\,d)$
Rule-6: Risk Mitigation
$iot{:}\,CriticalDevice(?\,d) \wedge iot{:}\,hasVulnerability$
$(?\,d, ?\,v) \wedge iot{:}\,mitigatedBy(?\,v, ?\,m1) \wedge$
$nrl{:}\,SecurityObjective(?\,m1) \wedge nrl{:}\,Security\_$
$Concept(?\,sc) \wedge nrl{:}\,supportsSecurityObjective$
$(?\,sc, ?\,m2) \wedge owl{:}\,SameAs(?\,m1, ?\,m2)$
$\Rightarrow\ iot{:}\,candidateSecurityCapability(?\,d, ?\,sc)$

Listing 1: Inherent Constraints

We established these relationships by modeling a variety of inherent constraints. Some examples of such constraints are given in *Listing 1*. With regards to operational dependencies, the related devices are linked using the *iot:hasDependency* object property. A sample constraint is given as Rule-1 for controller dependency. A given IoT device is dependent on its designated controller(s), which regulate its operations by issuing appropriate commands. Similar dependency relationships can also be established for other types of IoT entities, while catering for their interaction requirements. Inherent constraints are also used for pairing the devices with reference to their matching security properties. A sample constraint is given as Rule-2. This rule checks for authentication pairing at application / service level by utilizing nrl:ServiceSecurity ontology, which is already aligned with OWL-S ontology. As shown in Rule-2, authentication pairing is defined by using nrl:securityRequirement and nrl:securityCapability properties, which are registered as sub-properties of OWL-S:serviceParameter.

Rule-1: Controller Dependency
$ssn{:}Device(?d1) \wedge iot{:}hasRole(?d1, ?c) \wedge$
$iot{:}Controller(?c) \wedge iot{:}controls(?c, ?d2) \wedge$
$ssn{:}Device(?d2) \Rightarrow iot{:}hasDependency(?d2, ?d1)$
Rule-2: Authentication Pairing
$ssn{:}Device(?d1)$
$\wedge\ iot{:}hostsApplication(?d1, ?a1)$
$\wedge\ nrl{:}SecurityRequirement(?a1, ?r1)$
$\wedge\ nrl{:}AuthenticationAlgorithm(?r1)$
$\wedge\ ssn{:}Device(?d2)$
$\wedge\ iot{:}hostsApplication(?d2, ?a2)$
$\wedge\ nrl{:}SecurityCapability(?a2, ?c1)$

Listing 2: Risk-Driven Constraints

Risk-Driven Constraints

We utilized the power of deductive reasoning to analyze risks for a given IoT system. This knowledge was then used in conjunction with inherent constraints to isolate the required security capabilities, which can be used to mitigate these risks. As mentioned earlier, we considered the domain of smart home automation system and leveraged the empirical risk analysis study conducted by Jacobsson et al. [16].

Listing 2 presents some of the rules supporting this inference process. Rule-3 proposes an automated means of identifying vulnerabilities, induced due to the lack of security inter-operability among dependent devices. The rule states that if a given pair of dependent devices are deprived of encryption pairing then they are vulnerable to *poor_confidentiality*. Since SWRL is based on open-world assumption, execution of Rule-3 requires the world to be closed using suitable axioms. Rule-4 consumes the vulnerability and threat information associated with respective devices to infer the types of risks applicable on them. Next, Rule-5 categorizes the IoT assets with regards to the level of risk exposure. It isolates the devices with risk scores greater than or equal to 10 (in accordance with the categorization made by [16]), as members of *iot:CriticalDevice*. Finally, Rule-6 recommends the list of those security capabilities, which can be used to mitigate the vulnerabilities inducing that risk. These capabilities are linked with the device under risk using *candidateSecurityCapability* property.

Querying the Inferred Knowledge

After reasoning over the registered IoT data using OWL restrictions and rule-based constraints, information regarding the desired set of security capabilities can be inferred. This information can further be scrutinized through customized queries. For instance, queries can be used to isolate those dependent devices, which cannot communicate using an inter-operable security protocol or do not meet the desired security objectives.

Similarly for risk mitigation, only those capabilities can be selected, which are also supported by the dependent devices, as demonstrated by the query given in *Listing 3*. The query generates two sets, *S1* for capabilities required by each critical device for risk mitigation and *S2* for capabilities supported by its respective dependent devices. It then enlists the common capabilities by using the *sqwrl:intersection* built-in set operator.

Implementation and Evaluation

We have used **Protégé** software [20] to build and extend our ontology. Protégé is a W3C standard-compliant, free and open-source ontology-editor tool developed and maintained by Stanford University. Ontology inference and consistency checking was performed by utilizing the **Pellet**-engine, which is an OWL2, java-based, open source reasoner and comes pre-configured with Protégé. Pellet can not only be used to perform traditional reasoning tasks such as classification, debugging and querying with soundness and completeness, it additionally allows the use of SWRL and SQWRL built-ins in the rules, facilitates incremental classification and also supports reasoning through Jena in addition to OWL API interface.

$iot{:}CriticalDevice(?c) \wedge\ ssn{:}Device(?d)$
$\wedge\ iot{:}hasDependencey(?c, ?d)$
$\wedge\ iot{:}candidateSeurityCapability(?c, ?sec1)$
$\wedge\ nrl{:}hasSeurityCapability(?d, ?sec2)$
$\circ\ sqwrl{:}makeSet(?S1, ?sec1)$
$\wedge\ sqwrl{:}groupBy$
$(?S1, ?c) \wedge sqwrl{:}makeSet(?S2, ?sec2)$
$\wedge\ sqwrl{:}groupBy(?S2, ?d)$
$\wedge\ sqwrl{:}intersection(?S3, ?S1, ?S2)$

Listing 3: Report Matching Security Capabilities

For IoT systems, we conducted an extensive survey of the real-world IoT devices and their capabilities, mainly targeting the domains of home automation and building management systems. Risk profiles for these devices were built by leveraging the risk categories and corresponding likelihood values of our reference study [16]. However, values for risk impact were intuitively assigned, while catering for the device operational goals and capabilities. For example, impact of a security breach on a smart door lock or a smoke sensor will be considerably large as compared to a smart light, since the former can threaten the physical security and safety of the premises respectively. Contrary to that, for a given risk, the referred study [16]assigned same impact values to the complete group of IoT devices. The IoT device-specific information was extracted from openly available resources and was subsequently structured using an H2 database engine (a Java SQL RDBMS). Information from the database was mapped and populated as ontology instances using the ontopPro [21] data import plugin. OntopPro is a DB-ontology mapping editor plugin for Protégé, which offers a powerful and intuitive mapping language to generate RDF triples (ABox assertions) for the targeted ontology. Additionally, it also supports querying the database on the fly without the need to import it in the ontology.

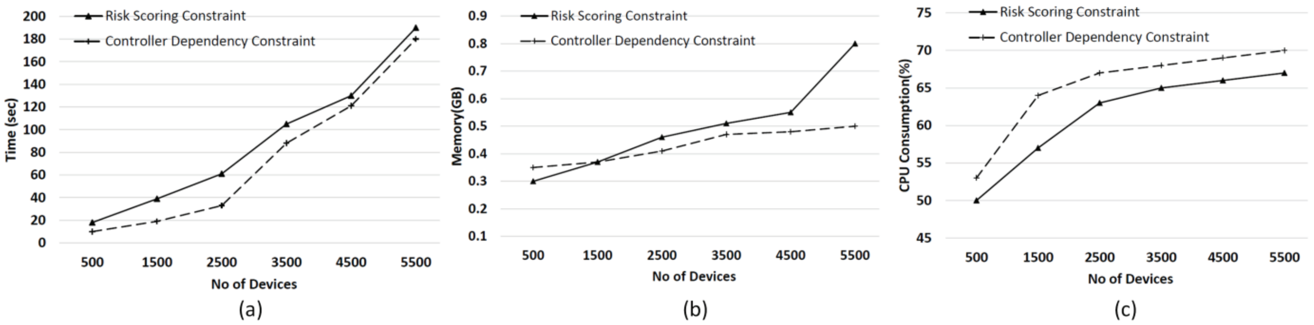

Fig. 3: (a) Impact of network size on constraint verification time (b) Impact of network size on memory requirements (c) Impact of network size on CPU load.

We developed a number of SWRL/SQWRL constraints such as for mapping IoT dependencies, aligning their security properties, identifying and quantifying associated risks and mapping applicable security controls for critical devices. OWL restrictions and SWRL constraints were processed by the reasoner to derive and update the inferred information. Subsequently, SQWRL based queries were run to output risk reports alongside suitable recommendations. We tested our system for accuracy and scalability as discussed below:

Accuracy: The accuracy of the system was verified through different ways. First of all, the correctness of the inferred knowledgebase was stamped by the reasoning engine, which ensured the soundness and completeness of the results through inherent features. Soundness preserves the accuracy by ensuring that all inferences based upon facts are valid with reference to the semantics. Conversely, completeness ensures that knowledge that is actually true can be correctly and completely inferred by the system. We also verified the accuracy with the help of a ground truth scenario based on a small-scale home-automation system. Different security constraints were tested by firing the SWRL based rules on the registered IoT facts and then manually comparing the inferred configurations with the facts, thus verifying the correctness.

Scalability: With regards to scalability, we fed IoT data of varying size into the ontology and analyzed the time and space performance of verifying different constraints. We utilized a corei5 machine with a 4GB RAM for the experiments. The results of experiments are given in Fig. 3, plotting time, memory and processing power consumed in relation with the number of registered devices. The results reflected a near-linear rise in time and memory requirements while increasing the network size.

Conclusion

In this paper, we presented an ontology-driven approach for risk analysis of IoT systems. The methodology adopts an automated way of semantically registering the security and functional properties of IoT elements, pairs these elements based on their respective parameters and subsequently verify their correctness in relation to desirable security configurations, applied as constraints over the registered information. We are actively working towards extending this work in several directions. The IoT and security ontologies presented in this paper are being extended to comprehensively model related IoT aspects such as behavior, operational specifications and network topologies. We also plan to port our ontology into Apache Jena [22], a free and open source Java-based framework for linked data applications. Our initial experiments over Jena have shown much improved results in terms of scalability, data materialization time and reasoning efficiency.

Acknowledgements

A part of this research was supported by Higher Education Commission (HEC), Pakistan through *Indigenous 5000 PhD Fellowship program (Phase-II).*

REFERENCES

1. H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelffl´e, Eds., Vision and Challenges for Realising the Internet of Things. Luxembourg: Publications Office of the European Union, 2010.
2. P. Guillemin, P. Friess et al., "Internet of things strategic research roadmap," The Cluster of European Research Projects, Tech. Rep., September, 2009.
3. D. L. McGuinness, F. Van Harmelen et al., "OWL web ontology language overview," W3C recommendation, vol. 10, no. 10, 2004.
4. I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean et al., "SWRL: A semantic web rule language combining OWL and RuleML," W3C Member submission, vol. 21, p. 79, 2004.
5. M. J. O'Connor and A. K. Das, "SQWRL: A Query Language for OWL," in OWLED, vol. 529, 2009.
6. M. Compton, H. Neuhaus, K. Taylor, and K.-N. Tran, "Reasoning about sensors and compositions." in SSN. Citeseer, pp. 33–48, 2009.
7. M. Calder, R. A. Morris, and F. Peri, "Machine reasoning about anomalous sensor data," Ecological Informatics, vol. 5, no. 1, pp. 9–18, 2010. [Online]. Available: http://linkinghub. elsevier.com/retrieve/pii/1574954109000715.
8. L. Lefort, C. Henson, K. Taylor, P. Barnaghi, M. Compton, O. Corcho, R. Garcia-Castro, J. Graybeal, A. Herzog, K. Janowicz et al., "Semantic Sensor

Network XG-final report," W3C Incubator Group Report, vol. 28, 2011.

9. M. Bermudez-Edo, T. Elsaleh, P. Barnaghi, and K. Taylor, "IoT-Lite Ontology," https://www.w3.org/Submission/2015/SUBM-iot-lite-20151126/, accessed: 01-08 -2016.

10. S. De, P. Barnaghi, M. Bauer, and S. Meissner, "Service modelling for the internet of things," in Federated Conference on Computer Science and Information Systems (FedCSIS). IEEE, pp. 949–955, 2011.

11. M. B. Alaya, S. Medjiah, T. Monteil, and K. Drira, "Toward semantic interoperability in oneM2M architecture," IEEE Communications Magazine, vol. 53, no. 12, pp. 35–41, 2015.

12. A. Gyrard, C. Bonnet, and K. Boudaoud, "An ontology based approach for helping to secure the ETSI machine-to-machine architecture," in ITHINGS, September 1-3, 2014, Taipei, Taiwan, China, 09 2014. [Online] http://www.eurecom.fr/publication/4322.

13. C. Liu, Y. Zhang, J. Zeng, L. Peng, and R. Chen, "Research on dynamical security risk assessment for the internet of things inspired by immunology," in Eighth International Conference on Natural Computation (ICNC). IEEE, pp. 874–878, 2012.

14. R. Roman, P. Najera, and J. Lopez, "Securing the internet of things," Computer, vol. 44, no. 9, pp. 51–58, 2011.

15. S.-I. Chang, A. Huang, L.-M. Chang, and J.-C. Liao, "Risk factors of enterprise internal control: Governance refers to internet of things (iot) environment", RISK, 2016.

16. A. Jacobsson, M. Boldt, and B. Carlsson, "A risk analysis of a smart home automation system," Future Generation Computer Systems, vol. 56, pp. 719–733, 2016.

17. A. Gangemi, "DOLCE Ultralight Ontology," http://www.loa.istc.cnr.it/ontologies/DUL.owl, 2007, accessed: 2016-11-09.

18. A. Kim, J. Luo, and M. Kang, Security ontology for annotating resources. Springer, 2005.

19. D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne et al., "OWL-S: Semantic markup for web services," W3C submission, vol. 22, pp. 2007–04, 2004.

20. H. Knublauch, R. W. Fergerson, N. F. Noy, and M. A. Musen, "The Protégé OWL plugin: An open development environment for semantic web applications," in The Semantic Web–ISWC, Springer, pp. 229–243, 2004.

21. KRDB Research Group, "ontopPro: The OBDA Plugin for Protégé," http://ontop.inf.unibz.it/,accessed: 2016-11-09.

22. A. Jena, "A free and open source java framework for building semantic web and linked data applications," URL: http://jena.apache. org, 2011.