A Semantic Model for Cyber Security

Bruce Barnett
Andrew Crapo

GE Global Research Center
1 Research Center
Schenectady, NY 12309

barnettbr@ge.com
crapo@ge.com

Keywords: CyberSecurity, SmartGrid, Ontology, Sematic Web

Abstract

Smart Grid security is challenging as experts in both IT Security and ICS [Industrial Control Systems] systems are few. Expertise in multiple domains is needed, and tools that can be used to analyze smart grid systems during the design phase are non-existent. We used Semantic Web Technology to create an ontology that is capable of reasoning about security attributes. We used SADL (Semantic Application Design Language) to describe information about a smart grid system such as network topology, device specifications, and site-specific information. SADL uses English-like statements to build a model that is understandable by domain experts without requiring knowledge of Semantic Web technology.

We describe components of the ontology that are capable of describing, measuring, and comparing both physical and network-specific risks. We also developed a GUI that displayed the results in a Failure Mode Effects Analysis, where threats are prioritized by Likelihood, Detectability, and Severity.

1. INTRODUCTION

Security is a difficult problem in any domain, as it requires expert knowledge from multiple domains - more than any single Subject Matter Expert can master. To elaborate, security expertise can be subdivided into several categories including:

- Physical Security (buildings, fences, locks, alarms)
- Device security (requires specialized knowledge of the device)
- Reverse Engineering (ability to extract and duplicate proprietary algorithms and code from compiled programs)
- Network control security (requires knowledge of network topology and protections)
- Application security (requires knowledge of the SCADA software accessing a device)
- Network Protocol security (requires understanding of protocols and packet contents)
- Operational security (requires knowledge of the overall system requirements and functionality).

When considering the SmartGrid and SCADA/ICS (*Supervisory Control And Data Acquisition/Industrial Control Systems)*, a merged understanding of these issues is essential, and finding experts is problematic. Therefore, having tools that can assist in this process is desirable. Ideally, these tools should have the following capabilities:

- They should be able to measure system security in an objective and repeatable fashion.
- They should allow information from multiple domains of knowledge to be merged.
- They should be able to capture and re-use knowledge and expertise from experts.

When we started the project, we felt that Semantic Web technology could be used to address this problem. We proceeded to build a prototype to determine the suitability of the technology.

2. WHAT IS THE SEMANTIC WEB?

Communication will always be imprecise when there is disagreement over the meaning of words. The word “network”, for instance, has different meanings to companies such as NBC, Cisco, and Facebook. It is

therefore necessary to formally define the words that will be used, along with their attributes and relationships with other words, to allow programs to analyze systems based on these words.

The World Wide Web provides syntactically correct information to a web browser, which has the task of rendering the data. However, complex problems either require humans to interpret the data, or specially crafted scanners to interpret web pages.

The Semantic Web is a technology that (a) defines an ontology, (b) provides meaningful data using this ontology over the web, and (c) provides rules that use this data, and (d) allows semantic reasoners to deduce information using the rules and data.

3. SEMANTIC WEB STANDARDS

The Web Ontology Language, OWL[1], is a formal (unambiguous) language that builds on RDF and RDFS and is informed by prior work including Description Logics (DL), OIL, DAML, and DAML+OIL. The Semantic Web Rule Language (SWRL)[2], a W3C proposal that combines OWL (DL and Lite) with a subset of the Rule Markup Language (RuleML), or the Jena[3] language can be used to express rules that semantic reasoners such as Jena, or Pellet[4] can use for reasoning. The results can be queried using SPARQL (RDF query language)[5].

However, developing tools based on OWL, SWRL and SPARQL is difficult. OWL is designed for computer parsing systems, not for easy consumption by people. Simlarly SWRL syntax is not English-like and is difficult to understand. If an ontology is modified in an OWL file, the corresponding change in a SPARQL query is not obvious and may not be immediately identified.

4. USING SADL TO DEVELOP SEMANTIC WEB APPLICATIONS

SADL[6], which stands for Semantic Application Design Language, is an open-source solution, developed by GE that addresses the above problems. It allows ontologies to be expressed in an English-like language. Ontologies written in SADL are automatically converted into OWL. Rules written in SADL can be converted to Jena rules or SWRL, allowing the user in the Eclipse environment to launch and query semantic reasoners, such as Jena and Pellet. Ontology modifications that break existing rules and models are immediately identified and can be addresses without requiring any queries. In other words, rules and ontologies can be easily modified and reasoning solutions can be quickly developed.

The use of SADL to model Smart Grid systems has been previously demonstrated.[7][8]. We decided to use SADL to both specify an ontology and to specify rules that can calculate the security of a design.

5. SEPARATION OF KNOWLEDGE DOMAINS

We created an ontology that separated certain domains, allowing each to provide input independently. These domains include

- Device Library – allowing the specification of device characteristics independent from configuration-specific information. Each device type was a unique class.
- Physical topology – allowing the generation of network topology information from a network layout design package.
- Network Protection – specifying the controls that protect and isolate a network, using information provided by the network operations department.

We divided the threats into physical and network-related threats to test the usefulness of the technology.

6. PHYSICAL SECURITY

We created the following ontology to describe attributes necessary to calculate physical security. Our base class is **Device**, The SADL specification is

Device is a top-level class, described by **physicalPenetrationEffort** with a single value of type float, described by **tamperDetectionProbability** with a single value of type float.

We created a generalization of the base class called **Container**:

Container is a type of **Device**, described by **containsDirectly** with values of type **Container**.

Containers are used to describe physical barriers (fences, buildings) as well as electronic cabinets, racks, cases used to house electronics. Any device may have the ability to detect tampering in the model. As an example, the protection for a power relay, and the layers of protection for this device, and the values of the attribute, are summarized in Table 1. In our ontology, the outermost containers can be identified because they are not contained inside of other containers. We created rules to identify these items and calculate the total hours to penetrate all of the physical defenses and expose the innermost device. Using the data in Table 1, the total effort would be 14 hours. This value, multiplied by **AttackerSkill**, was used to calculate an estimated time to physical compromise the corresponding device. For example, an Expert (value of 0.5) could breech the physical defenses in 7 hours, and a Novice (value of 2.0) would take 28 hours.

To calculate the probability of detection, each of the physical barriers optionally had a tamper detection probability. We used Equation 1 to calculate the total Detection Probability (DP). In the example in Table 1, the chassis and the building have alarms, with probability of the alarm detecting tampering at 80% and 90% respectively.

Table 1. Example of Physical Protection by Containers

Table of Physical Protection Characteristics		
Device	Physical Penetration Effort (hours)	Tamper Detection Probability
Relay Electronics	2	0
Relay Case	1	0
Chassis	1	80%
Building	10	90%
Fence	3	0

Equation 1. Detection Probability

$$\sum DP = 1 - (1\text{-}DP(\text{Layer}_1) * (1\text{-}DP(\text{Layer}_2)) \ldots * (1\text{-}DP(\text{Layer}_n))$$

Using the data in Equation 1, the overall probability of detection would be 1- (1-.90)*(1-.80) = .98 (98%). We did not modify the detection probability based on **AttackerSkill.** Better models for attacker skillsets are, or course, possible.

We added a **numberOfHomesServicing** attribute that specified the number of homes that would be impacted if the device was compromised. The **controlOf** attribute was used to describe when one device has supervisory control over another device. The cumulative number of homes under control of a single device could then be calculated using rules that total up the number of homes under the control of a single device, both directly, and indirectly (when a compromised device has control of a non-compromised device).

These specified and derived attributes allows us to calculate for any device, the detectability, likelihood, and severity of a physical compromise, in a modified Failure Mode Effects Analysis (FMEA) formula. We wrote a Java client that queried the reasoning engine, and extracts this information. To allow comparison, we normalized the raw values, converting them to a value from 1 to 10, using a scale appropriate for each measurement. A higher number corresponded to a greater impact of the device's compromise. The three values are multiplied to create a Risk Prioritization Number (RPN). By sorting on the RPN, we could identify devices that have the greatest impact if compromised. A screenshot of this tool is shown in Figure 2.

7. NETWORK CONNECTIVITY

The ontology to describe network connectivity is shown in the following SADL syntax. We first created a class corresponding to Layer 1 of the ISO mode, i.e. **PhysicalLinkLayer**. We also created two generalizations of the class Device. **ComDevice** describes a **Device** with a network connection:

- **ComDevice** is a type of **Device**, described by **attachedTo** with values of type **PhysicalLinkLayer,** described by **hasMACAccessTo** with values of type **ComDevice**, described by **hasIPAccessTo** with values of type **ComDevice.**
- **Network Controller** describes a device that connects 2 or more networks together, providing connectivity between two or more **PhysicalLinkLayer**networks:

NetworkController is a type of **ComDevice**, described by **providesMACBridging** with a single value of type boolean, described by **providesIPRouting** with a single value of type boolean.

The two Boolean attributes, **providesMACBridging** and **providesIPRouting** are used to describe switches, hubs, routers and firewalls. We wrote rules that derive MAC-level and IP-Level connectivity based on these attributes, allowing us to assign values (**Devices**) to **hasMACAccessTo** and **hasIPAccessTo,** attributes, which can be used to determine connectivity between any two devices.

8. NETWORK-BASED ATTACKS

In a preliminary version of our model, we wrote explicit rules that deduced network-based attacks. This technique became very unwieldy, as the attacks were complicated, created many specific attributes, and rule interaction grew as the number of attacks grew.

We replaced this approach with a generalized attack/defense ontology (Figure 1) that has several advantages. The ontology describes device capabilities, attack mechanisms, vulnerabilities, and defense mechanisms as classes, and the specific attack or defense is modeled as an instance of the class.

Before an attack is possible, the device must have Device Characteristics that allow the attack. These characteristics could include required operating systems, the ability to promiscuously sniff traffic, the ability to have raw socket access, patches, access privileges, etc. A compromised device with these characteristics can attack another device using a specific Attack Capability. Each attack is associated

with a specific vulnerability associated with a device, and the target of the attack must have the vulnerability.

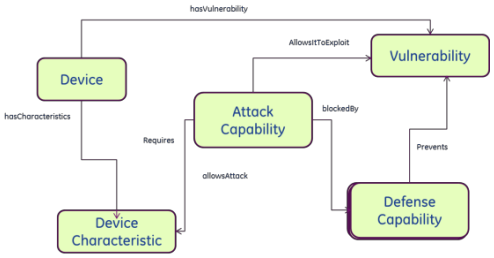


Figure 1. Attack/Defense Ontology

For each attack, there may or may not be a protection mechanism available. This could be host-based or network based. If the vulnerable host has the defensive capability, it can prevent that specific attack. If not, it could become compromised.

The defense capability can be network-based as well. For example, a firewall could protect a device from attack.

We created rules that determined if any **NetworkController** had the defensive capability for a particular vulnerability, and was also between the attacker and the target. If so, the target would not become compromised. Therefore the model is capable of determining which vulnerabilities pose a danger of compromise, resulting in the compromise of additional devices.

Using a SADL-based prototype based on this ontology, we created an instance of a StuxNet-like virus. The modeling environment can determine which devices lack protection, and are therefore vulnerable. This allowed us to perform "what-if" analysis, and by calculating the total **numberOfHomesServicing** for all devices either directly compromised, or under control of a compromised device, we are able to calculate the Defense-in-Depth for each device in the network.

9. SUMMARY AND CONCLUSIONS

We believe that the two prototypes we built demonstrate that semantic reasoners can provide useful benefit in measuring the overall security of a complex network. We met the desired goals with the described technology by (a) measuring the security of a complex system both objectively and repeatably; (b) merging information from multiple domains of expertise; and (c) being able to capture and re-use knowledge of experts.

When developing ontology-based solutions, SADL provides both flexibility and a rapid prototyping environment for developing new ontologies. In our ontological model, only a few attributes and classes are needed to calculate metrics related to physical and network security.

We also believe that the ontology presented is simple and practical, and can be extended to more sophisticated applications. It allows expression of information from multiple domains, and therefore can be used to combine the expertise of multiple experts. In addition, knowledge can be incrementally added to this model, improving the sophistication and capability of the model as needed. We find that the proper ontology provides a framework for multiple applications to share and exchange information.

References

[1] "OWL Web Ontology Language Reference", February 10, 2004, Mike Dean and Guus Schreiber, eds. http://www.w3.org/TR/owl-ref/

[2] "SWRL: A Semantic Web Rule Language Combining OWL and RuleML", May 21, 2004, Ian Horrocks et al., http://www.w3.org/Submission/SWRL/

[3] Jena – A Semantic Web Framework for Java. http://jena.sourceforge.net/

[4] Pellet: OWL 2 Reasoner for Java. http://clarkparsia.com/pellet/

[5] SPARQL Query Language for RDF. http://www.w3.org/TR/rdf-sparql-query/

[6] SADL SoureForge Home http://sadl.sourceforge.net/

[7] Andrew Crapo, Xiaofeng Wang, John Lizzi, and Ron Larson. *The Semantically Enabled Smart Grid* (2009). Grid Interop 2009. Available at http://www.gridwiseac.org/pdfs/forum_papers09/crapo.pdf

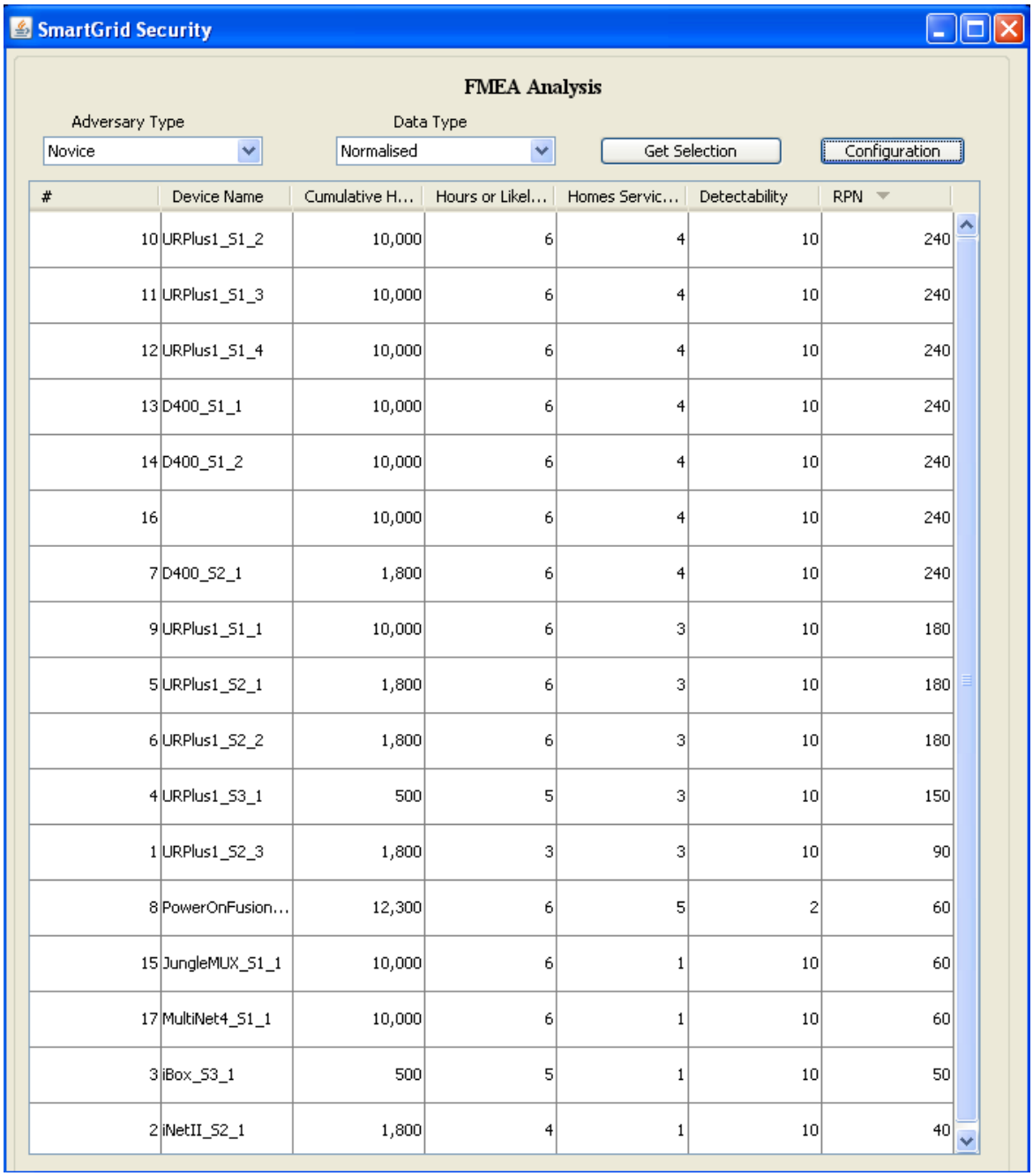
[8] Andrew Crapo, Kathrina Griffith, Ankesh Khandelwal, John Lizzi, Abha Moitra, Xiaofeng Wang, "Overcoming Challenges Using the CIM as a Semantic Model for Energy Applications", Grid-Interop 2010. http://www.pointview.com/data/files/3/2433/1730.pdf

Biography

Bruce Barnett graduated with a BS is Mathematics of Computation from RPI in 1973. Bruce was the primary Software Engineer and for 12 tears developed and maintained the major product line of the Factron division of Schlumberger. In 1988, Bruce joined GE's Global Research Center in Niskayuna, NY. Bruce developed working prototypes and published 21 papers on secure wireless sensor protocols (2005), complexity-based intrusion detection system (2002-2009), secure data provenance (2009), security vulnerability (2000) and expert-system-based fault analysis programs (1995) using GEN-X with Andy.

Andrew **Crapo** received a B.S. in Physics from Brigham Young University in 1975, an M.S. in Energy Systems from

the University of Central Florida in 1980, and a Ph.D. in Decision Sciences and Engineering Systems from Rensselaer Polytechnic Institute in 2002. He is a senior professional information scientist at the GE Global Research Center where he has worked since 1980. His work has focused on applications of information science to engineering problems including applied artificial intelligence, human-computer interactions, and information system architectures. More recently he has focused on modeling and the application of Semantic Web technologies to engineering and business problems.



#	Device Name	Cumulative H...	Hours or Likel...	Homes Servic...	Detectability	RPN
10	URPlus1_S1_2	10,000	6	4	10	240
11	URPlus1_S1_3	10,000	6	4	10	240
12	URPlus1_S1_4	10,000	6	4	10	240
13	D400_S1_1	10,000	6	4	10	240
14	D400_S1_2	10,000	6	4	10	240
16		10,000	6	4	10	240
7	D400_S2_1	1,800	6	4	10	240
9	URPlus1_S1_1	10,000	6	3	10	180
5	URPlus1_S2_1	1,800	6	3	10	180
6	URPlus1_S2_2	1,800	6	3	10	180
4	URPlus1_S3_1	500	5	3	10	150
1	URPlus1_S2_3	1,800	3	3	10	90
8	PowerOnFusion...	12,300	6	5	2	60
15	JungleMUX_S1_1	10,000	6	1	10	60
17	MultiNet4_S1_1	10,000	6	1	10	60
3	iBox_S3_1	500	5	1	10	50
2	iNetII_S2_1	1,800	4	1	10	40

Figure 2. Screen Capture of FMEA Tool