Reduced Realistic Attack Plan Surface for Identification of Prioritized Attack Goals

Jeffrey Smith, PhD
Decision Aids and Planning Directorate
BAE Systems
Burlington, MA
jeffrey.smith5@comcast.net

Michael Figueroa, CISSP
Intelligent Adaptive Software Directorate
BAE Systems
Burlington, MA
michael.figueroa@baesystems.com

Abstract— **Current homeland cyber security practices and techniques focus on identifying weakness, aggregating data in the hope of improving incident detection and promoting information sharing with the public. These foci emphasize a broad collection of knowledge of what could happen to minimize damage and urge the implementation of a frequently growing list of specific actions to reduce vulnerability exposures. Large-scale "big data" extraction and processing challenges ensue from the need to understand each device vulnerability within the context of every environment. Conversely, an attacker needs only to identify a single attack vector to elicit a compromise**. **Our paper describes 1) a more efficient approach to proactively securing devices using an Attack Plan Generator, 2) how we transform vulnerability and defect databases into attack surface representations and 3) how those representations provide a much more effective perspective into how an attacker would seek to compromise a given device.**

Keywords-cyber, attack surface, vulnerability, attack planning, adversary goals, semantic alignment.

I. INTRODUCTION

Current homeland cyber security practices and techniques focus on identifying weakness (US CERT), aggregating data in the hope of improving incident detection (EINSTEIN[1]) and promoting information sharing with the public. These foci 1) emphasize a broad collection of knowledge of what could happen to minimize damage and 2) urge the implementation of a frequently growing list of specific actions, usually against a single device, to reduce vulnerability exposures. Large-scale "big data" extraction and processing ensue from the need to understand each device vulnerability within the context of every environment. Conversely, an attacker need only identify a single attack vector to elicit a compromise.

For a simple example of how the scope differs between an academic vulnerability survey and an attack-centered assessment, we can use a modern network router. That device may expose 10 functions for external connections and management. Each of those functions may connect to a distinct software module (i.e. SSH, Telnet, Management Interface, Web Server, etc.). If each module has just 10 capabilities or vulnerabilities that an attacker could use to gain greater access to the environment that network router operates in, then a general attack survey that lacks the context of how the functions work together would result in an attack surface with 100 separate *potential* exploit points. As most devices would use a series of methods to connect (e.g. connect to the Management Interface over SSH), then the interactions between services may represent more potential exploit points, extending the attack survey exponentially. An attack-centered assessment would highlight only those features and vulnerabilities that could represent *plausible* exploit points (e.g. selecting out (excluding from analysis) using the Web Server over Telnet as an implausible combination of services).

Any reasonably complex device could be expected to be at risk of a substantially large set of *potential* unique attack scenarios, given moderate length attack plans and a number of ways to achieve the goal of each set. Our objective is to establish the basis for a smarter defense posture founded on a much smaller subset of *plausible* attack scenarios.

Contrasting the vulnerability-focused and attack-centric cyber security positions illustrates how cyber-threat databases are insufficient for securing computational devices because they describe vulnerabilities from a computer science research view rather than from that of a creative, inventive attacker. Whereas the former emphasizes a methodical and inclusive examination that would likely represent a large-scale data analysis challenge to cull valuable but uncommon attack-related information from a much broader sample of noise, the latter simplifies the exercise by combining knowledge, context, and data to eliminate much of the noise. Compromising sensitive assets requires much more than identifying vulnerabilities. The attacker also considers a device's properties, operational context and potential attack exposure points with respect to specific exploitation goals. It's the characterization and modeling of the broad environment in which a device functions that presents an advantage to an attacker - seeking to compromise an information asset - over a defender that focuses on the vulnerabilities embodied by discrete devices.

We represent a device-under-test with a probabilistic weighted set of candidate services and configurations. This device-under-test information is combined with vulnerability information extracted from multiple cyber threat-related sources, resulting in a complex attack surface. We show how one can prioritize candidate device configurations and known

[1] US-CERT is United States Computer Emergency Readiness Team at http://www.us-cert.gov. US-CERT Einstein Program provides an automated process for collecting, correlating, analyzing, and sharing computer security information across the Federal Government to improve our nation's situational awareness.

Approved for Public Release; Distribution Unlimited. Cleared for Open Publication on 8/19/2013.

vulnerabilities with heuristics to simplify the attack surface that is most likely to be exploited. The reduced "critical" attack surface is used to identify a structured, prioritized attack goal set based on the security goals of the user. These attack goals consist of subgoals, ordered by likelihood of applicability and risk significance.

Fig. 1 depicts the Knowledge Acquisition (Section II) and Attack Plan Generator (Section III) components used to generate a prioritized checklist of software/firmware and classes of malicious functionality to rule out.

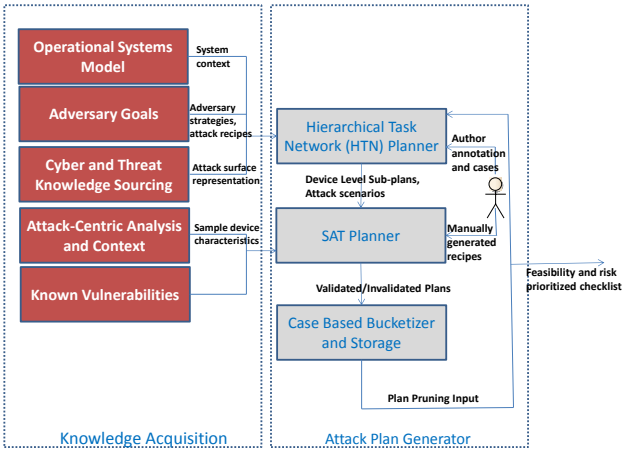

Figure 1. Knowledge Acquisition and Attack Plan Generator

II. KNOWLEDGE ACQUISITION

A. *Cyber and Threat Knowledge Sourcing*

Knowledge acquisition starts with identifying all of the data sources that may contain information that we would consider useful for representing the device attack surface. The National Vulnerabilities Database (NVD) [6] is a central repository that is highly structured and will serve as a key component of our knowledge acquisition activities. There are also other sources of information that we can leverage together with the NVD to refine our vulnerability definition and enhance how we identify, classify, and rate vulnerabilities and their associated attacks. Jarzombek [2] describes information and data sources that support our vulnerability analysis. We expect to supplement vulnerability information with other structured sources (i.e. US-CERT, OSVDB, vendor feeds) and unstructured sources (i.e. Exploit DB, BugTraq, online forums) conducting information quality assessments to identify the best data combination.

The critical challenges are to: 1) develop a consolidated attack-centered data repository from several relevant external data sources that contain varied content organization and restructure the data into our planning solution without modifying the structure for each source and 2) minimize the level of human contact needed to incorporate disparate data sources into the repository by promoting an automated solution set.

As detailed in [7], BAE Systems has developed an innovative approach to semantic alignment that is able to handle the intricacies of real-world data. Translation from source data into a semantically equivalent representation in a target ontology faces a number of problems, including structural dissimilarities (non-isomorphism) in the source and target data models, varied representations of data types, and disparities in the way properties and attributes are packaged into objects.

Leveraging our research on a mediation solution [7] that takes source data and a declarative mapping form source schema to the target schema, we employ a Semantic Alignment Mediator to transform each external data source record into a standardized Attack and Exposure Point (AEP) record based on a new Attack Surface Ontology (ASO) to normalize the data. We develop the ASO to accomplish two novel objectives to: 1) translate vulnerabilities into potential AEPs and 2) capture new attack-centered data that may include exploits, configurations, and new ways to leverage legitimate device functionality.

As an example of this mediation process, the NVD contains all the data enumerated in Table I, including the CPE specifying IT platform-naming schemes based on a relational model. Kladilkar [5] reported on data migration from this relational model to a semantic model using conversion applications and ontology and inference APIs to support RDF parsing and query. This approach is similar to [3][4][7] who used Protégé to build up OWL-DL ontologies and OWL-S as a framework to register service definitions and allow composite system definitions to evolve. We leverage this past research to normalize vulnerability and support its translation into a comprehensive attack-centric repository.

TABLE I. VULNERABILITY INFORMATION EXAMPLE TYPES AND DATA SOURCES

What the information provides	Location of information
What IT systems do I have in my enterprise?	CPE (Platforms)
What known vulnerabilities do I need to worry about?	CVE (Vulnerabilities)
What vulnerabilities do I need to worry about now?	CVSS (Scoring System)
How can I configure my systems more securely?	CCE (Configurations)
How do I define a policy of secure configurations?	XCCDF (Configuration Checklists)
What weaknesses could be exploited?	CWE (Weaknesses)
What attacks can exploit which weaknesses?	CAPEC (Attack Patterns)

A comprehensive AEP repository is built through ASO development detailing how hackers may conduct malicious activities. One key aspect will be to apply the advanced reasoning and data mapping methods provided through our mediation solution to attach new semantic "Weakness" and "Effect" descriptions to each record. For example, an attacker could exploit an SNMP Service vulnerability (CVE-2013-1105) to have a Privilege Escalation effect that enables the hacker to conduct a View Device Configuration File action. Those new semantic descriptions are critical for developing a clear attack model for a given device to improve vulnerability alignment with device capabilities that function as expected as well as for an attacker may use to attack a given target. This

way, we enhance our attack surface perspective of a given device.

We automate the process of capturing NVD-based databases that we can reason from as these databases frequently change. We evolve this automation using the Security Content Automation Protocol (SCAP) with web services to transform device characterizations, configurations and vulnerabilities into a suitable to Attack Plan Generator (Section III).

B. *Attack-Centric Analysis and Context*

The weakness of the technology-centric perspective promoted by contemporary security researchers, and embodied by the NVD, is that it comes at the expense of understanding how a hacker discovers and exploits vulnerabilities. Using a standard bottom-up research methodology that emphasizes how device services fail in a way to benefit the hacker only provides perspective on what can go wrong. Hackers champion context, seeking instead to understand how a device functions in a top-down manner as an assistant to capture a target. In this perspective, vulnerabilities within a given device have a potential relationship with the target that the hacker may leverage to gain greater control. When serving as the central knowledge acquisition vehicle, the NVD demonstrates the result of this heavy bottom-up focus, organizing and categorizing vulnerabilities against products and devices rather than expressing how an individual vulnerability may relate to, and even anticipate, an attack objective.

To implement a new attack-centric analysis that examines how vulnerabilities and legitimate device services relate to attack objectives, we construct a new AEP device view that aligns with how an adversary seeks to expose the device's attack surface. A device simply represents a combination of services, configurations, functions, and vulnerabilities when viewed from an attack perspective. Fig. 2 illustrates an example of that attack perspective as an abstraction atop attack graphs that depicts multistage cyber-attacks [8]. Whereas that research focuses on the layers within a complex network environment, we extend the attack graph definition to represent device services and functions as discrete attack objects. Each of those architectural objects represents potential exposure points that are accessible from only the next subsequent layer, with externally available services residing on the outer-most architecture layer.

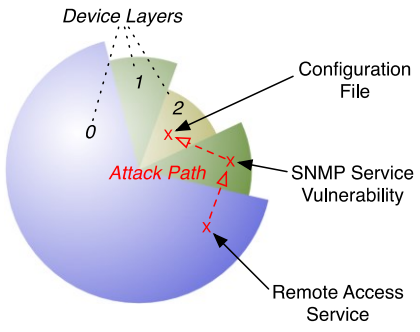

Figure 2. Network Device Attack Surface Example

Successful device service identification requires efficient gray-box reconnaissance, including probing, scanning, and static analysis techniques. Today, device reconnaissance and forensics is largely a cyber-art with no universal solution for every potential device-under-test. Rather than developing a new technique, we unify disparate techniques, such as those included in Table II, into an automated framework that populates a common device ontology for analysis by the planner.

TABLE II. FORENSIC AND ANALYTICAL TECHNIQUES FOR DEVICE CHARACTERIZATION

Action	Description	Example
Firmware Analysis	Identifies and extracts files, libraries and executable code embedded in firmware images based on "magic signatures."	Binwalk, firmware-mod-kit
Internal Scanning	Discovers attributes regarding internal device configuration and operation using standard utilities. Enumerates management services, device configuration, and hardware architecture.	ps, netstat, ifconfig, iptables, lspci, ls
External Probing	Characterizes OS and applications based on observable response to legitimate or malformed packets, TCP/IP timeout behavior, etc; compares to library of pathological response "signatures."	Nmap, RING, X-Probe, p0f

Our approach begins by applying contemporary automated forensic and analytical techniques to analyze the external and internal structures of a given device. The results allow us to then build a library of potential device functions that collectively represent a rational attack surface perspective of the complete device. When represented as a consistent Device Characterization Ontology, we may then establish linkages between device functions and AEP records to support attack plan exercises.

The next step to model potential adversary goals is conducting attacks against the device. Adversary goals may be described at a high level such as to take control of the device with administrative privileges, or with more detail such as an attack pattern (series of steps) leading to a specific software end state. We infer adversary goals from data collected on attacks that adversaries have already conducted against other systems based on attack patterns derived from the Common Attack Pattern Enumeration and Classification (CAPEC) [1]. Table III describes some examples of the ways CAPEC describes a device attack surface that we leverage for goal definition.

The process of identifying and prioritizing device vulnerabilities involves reasoning over many series of exploits that can rapidly lead to an undesirably huge set of states. Our approach uses the CAPEC attack patterns list as an organizing and filtering tool, providing two methods to reduce the search space: probability of attack pattern applicability and user interaction with the lists of attack categories and patterns.

User interaction involves selection of categories or attack patterns, either to include or exclude the selected data. If patterns or categories exist in a hierarchical tree, reduction or

inclusion occurs from the selected node and down the tree. Facilitating search with the possibility of human modification of any of the ASO structures used to generate attack goals and subgoals is important to 1) improve and direct search and test, 2) improve the goal generation process with spiked goals and plan fragments and 3) refine structures with learned behavior. This also facilitates parallel development and test, with any stage of our process aided with human constructed structures. In other words, we improve automated planning with human aided guidance.

TABLE III. A SAMPLE OF ATTACK CATEGORIES AND ATTACK PATTERNS FROM THE CAPEC DATASET

Some Categories	Some Attack Patterns
• Data Leakage Attacks Resource Depletion • Path Traversal • Injection (Injecting Control Plane content through the Data Plane) • Spoofing • Time and State Attacks • Abuse of Functionality • Functionality Misuse • Abuse of Communication Channels • Probabilistic Techniques	• Buffer Overflow via Environment Variables • Overflow Buffers • Server Side Include (SSI) Injection • Session Sidejacking • Clickjacking • Cross Zone Scripting • HTTP Request Splitting • Cross Site Scripting through Log Files • Cross Site Tracing • Command Line Execution through SQL Injection • Object Relational Mapping Injection

Fig. 3 illustrates how the Semantic Alignment Mediator supports our solution. In this example, to transform vulnerability data into attack−centric AEP records, we combine the output from the Attack Goals analysis and an applicable NVD XML entry. For the information to be useful, our solution needs to know that the SubGoal is the same as the "Effect" assigned to the NVD entry. To accomplish this unification, we make use of declarative mappings from the Attack Goals repository and NVD schemas to our common ASO ontology. The result is a set of RDF statements in the AEP knowledge

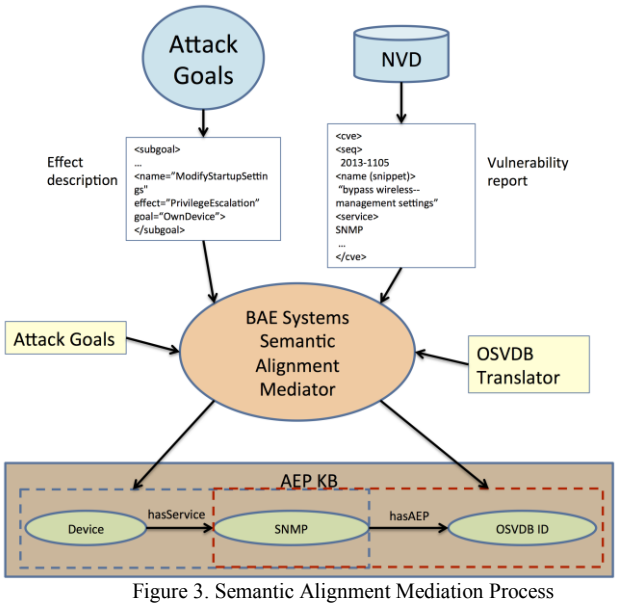

Figure 3. Semantic Alignment Mediation Process

base that links a device with a particular vulnerability that can be analyzed to confirm malice.

CAPEC attack patterns can also be used to seed the high-level planner portion of the Attack Plan Generator with attack plans or fragments of plans. Attack patterns have their own structure, with prerequisites, and these will be translated using the semantic alignment techniques discussed above into forms that the Attack Plan Generator (next section) can reason over.

III. ATTACK PLAN GENERATOR

As depicted in Fig. 1, we utilize a hybrid planning mechanism and multiple disparate sources of information to discover feasible attack plans. Alloy's (a SAT Solver described in [9]) capabilities for exhaustively evaluating hierarchically structured spaces of possible exploit techniques, combined with heuristic search to prioritize choices of device characteristics assumptions, allows a high level planner to generate high risk attack plan options. This approach utilizes a broad range of available information including known characteristics of the device being evaluated, uncertainties in device properties, repositories that support attack surface operators (e.g. CVE/Vulnerability [10]), the device class architecture and adversarial goals and strategies.

The planning engine supports checking for invalid pre-conditions of tactics. When constructing a plan, invalid preconditions can lead to backplanning to satisfy the preconditions, or to detect a conflict between the results in one subplan and the preconditions for another. If there is not enough data to determine whether a precondition is met, the appropriate analysis must be invoked. Thus, the tactics being considered shape the network analysis to be performed, and the results observed are folded into the cost calculations to direct the search. Our approach is a mixed initiative, using Alloy for enumerated elements, and mapping the Alloy representation to appropriate base classes.

Alloy takes the plan fragments and attack scenarios from the Hierarchical Task Network (HTN) planner and uses a Constraint Satisfaction Problem solver (Alloy) with heuristic constraint relaxation to create a minimally complete plan. The Activity Estimator uses the HTN's leaf nodes to create an initial Activity Graph, shown in Figure 4, which embodies at least one chain of causally ordered qualitative states that connect the start state and a goal state.

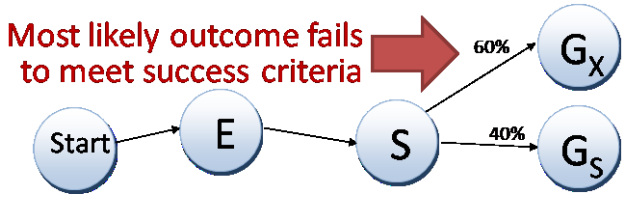

Figure 4. Shows a partial plan has a 60% chance of reaching Gx

Alloy generates every possible outcome for each individual activity, either validating the feasibility of the attack plan relative to its device model by fully instantiating plan, or refuting the plan by demonstrating no instantiation is possible. The combination of a heuristic search mechanism with Alloy enables reasoning from a diverse set of information and to

compensate for weaknesses the individual planners would be prone to if used in isolation. Search prioritizes seeking high risk candidates based on a combination of vulnerability related data drawn from general attack/adversary goals and known vulnerabilities of classes of devices and their services. Alloy uses a model of the device being assessed made specific through assumptions specified by heuristic search to exhaustively check for existence of a sequence of AEP-level [10] actions to achieve the adversary goals. (Note that the list of relevant AEPs is a function of the assumptions that constitute a case for Alloy consideration. For example, certain exploits are possible with one version of a service but not others.) This serves to answer the question "is there a way to perform the given actions even if the preconditions are false". (Formally, if the System Model is "Sys" and Properties are "P", Alloy tries to satisfy *Sys* ^ ⌐*P* using the built in constraint solver built into the Alloy Framework.)

An expanded form of the architecture, given in Figure 5, helps illuminate the process. Alloy uses heuristic search, along with a series of cases to evaluated, prioritized by estimated risk. This risk weighting over packages of assumptions combines

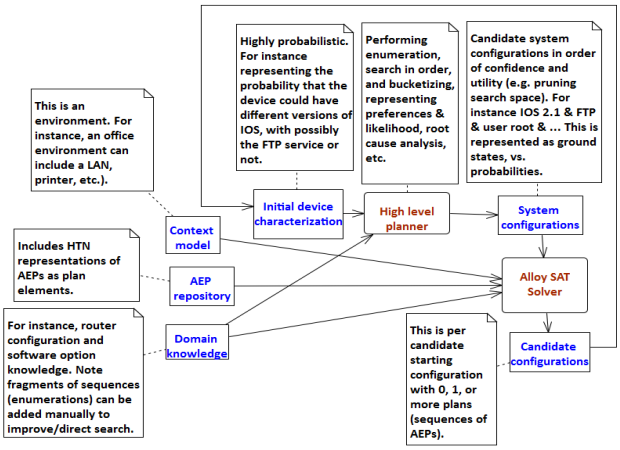

Figure 5. Hybrid Heuristic Planning Feeds Alloy's Modeling and Reasoning Capabilities to Efficiently Generate Feasible Attack Plans

the salience of alternative adversary goals with estimated probabilities of specific device characteristics combined with likelihood of discovering a valid attack plan. The calculation of the objective function for the heuristic search thus combines multiple sources of knowledge and data. The objective function itself is an estimate of risk and is a product of the salience of an attack and its estimated probability. Thus, the search mechanism seeks cases that maximize estimated risk: $R(c, g) = C(g)P(g|c)P(c)$ where:

- c - case or subgoal descriptor (in system configurations in Figure 5)
- g - the adversarial goal (in AEP repository in Figure 5)
- $C(g)$ - the level of damage that would result from the adversary achieving g (first estimated then learned)
- $P(g|c)$ - the probability of an adversary achieving g given assumptions c, which is equal to the probability of a viable attack plan to achieve g under conditions specified by c.
- $P(c)$ - The probability of the assumed device configuration, i.e. the combination of assumptions that specify a case. This probability is a product of the probabilities associated with the individual assumptions $P(c) = \prod_v P(v|c)$ where:
  - v indexes over the list of unknown service versions and other uncertainties regarding the detailed device properties. $P(v|c)$ is an expression of the frequency of specific versions and other attributes among the population of devices under consideration.

The probability associated with a given case can be understood as the product of a series of factors. We initially estimate $P(g|c)$ from probabilities associated with the portfolios of AEPs feasible given c: $P(g|c) = \prod_a P(a|c)$ where:

- a – enumerates the relevant AEPs given assumptions c
- $P(a|c)$ – is estimated from a combination of CVE related factors, including age, time, verbosity, author, and frequency of occurrence in the wild together with degree of similarity of the architectures on which this exploit was successful to the device characterization [3].

Factored together, the result is a master equation for risk weighting of cases:

$$R(c, g) = C(g) \prod_a P(a|c) \prod_v P(v|c)$$

These factors are available through table lookup, with entries derived from data sources or expert assessment, allowing the calculation of the objective function to be inexpensively calculated. Iterative evaluation of cases will generate new information regarding $P(a|c)$ whenever there has been a previous evaluation by Alloy of a case including the given AEP (indexed by a). An adaptive update of probabilities can be calculated using the usual Bayesian formula given a probabilistic distance estimate of the relevance of a previous case to the current one (bucketizing of results). Case-based reasoning methods provide one option for achieving this estimate.

Device characterization, AEPs, device class architecture, adversarial goals and strategies knowledge sources have been used as individual, but disparate, sources of vulnerability research. Our approach uniquely bridges all four sources with a planning approach that couples the best of constraint satisfaction, heuristic search and case-based reasoning based planning. We generate high likelihood of success plans first as a search product, and by this means will be able to cover a large and complex search space. The smaller search spaces presented to Alloy through this means will allow the constraint satisfaction based planner to comprehensively adjudicate the potential risk associated with possible device characteristics. Either way, this knowledge will be used to iteratively refine our search space by pruning parts of the network that are no longer known to be true or improving costs indirectly associated with action ordering or directly associated with vulnerability operator cost/weighting previously described.

IV. CONCLUSION

We have described a method to digest online and manufactured attack-related technical data to proactively secure devices using an Attack Plan Generator that combines a hierarchical task network planner to efficiently hypothesize likely attack scenarios with planning and formal model checking to prune infeasible attacks. While we are in the process of developing each of the related components, they are at varying levels of maturity (e.g. the Semantic Alignment Mediator [7] is the most mature), and we are identifying new development opportunities to connect the research "dots" to bring the Knowledge Acquisition and Attack Plan Generator vision into practice.

V. REFERENCES

[1] The MITRE Corporation, *Common Attack Pattern Enumeration and Classification*, http://capec.mitre.org .

[2] J. Jarzombek, *Software Assurance: Enabling Enterprise Resilience and Software Supply Chain Management, American Society for Quality (ASQ)*, http://www.acsac.org/2012/workshops/law/pdf/Jarzombek.pdf.

[3] J. A. Wang, H. Wang, M. Guo, L. Zhou and J. Camargo, *Ranking Attacks Based on Vulnerability Analysis*, Proceedings of the 43rd Hawaii International Conference on System Sciences - 2010.

[4] J. A. Wang and M. Guo, *OVM: An Ontology for Vulnerability Management*, CSIIRW 2009, April 13-15, Oak Ridge, Tennessee.

[5] V. Khadilkar, J. Rachapalli, B. Thuraisingham, *Semantic Web Implementation Scheme for National Vulnerability Database*, NIST Fifth Annual IT Security and Automation Conference, Baltimore, MD, 10/09.

[6] NIST, *National Vulnerability Database*, Version 2.2, http://nvd.nist.gov.

[7] BAE Systems, *Mediation, Alignment and Information systems for Semantic Interoperability (MAISSI): A Trade Study*, AFRL Technical Report, 2007.

[8] O. Sheyner and J. Wing, *Tools for Generating and Analyzing Attack Graphs*, LNCS 3188.

[9] D. Jackson, *Software Abstractions – Logic, Language and Analysis,* MIT Press, 2011.

[10] The MITRE Corporation, *Common Vulnerabilities and Exposures,* http://cve.mitre.org .