Applied Vulnerability Detection System

Jeffrey Smith, PhD
BAE Systems
Burlington, MA

Basil Krikeles, PhD
BAE Systems
Burlington, MA

David K. Wittenberg, PhD
BAE Systems
Burlington, MA

Mikael Taveniku, PhD
XCube Technologies
Nashua, NH

***Abstract*—In [1], we presented a Vulnerability Detection System (VDS) that can detect emergent vulnerabilities in complex Cyber Physical Systems (CPS). It used the attacker's point of view by collecting a target system's vulnerability information from varied sources, and populating a Attack Point (AP) database. From these APs, a Hierarchical Task Network generated the set of composite device-level attack scenarios. The VDS used Alloy [2] to reduce the cardinality of the generated space by evaluating the feasibility of each attack. This paper specializes prior research by submitting the generated prioritized list to an automotive-specific Attack Evaluation Process (AAEP). With a combination of simulation and vehicle instrumented real-time execution, the AAEP confirms each candidate attack. The AAEPs output is used as feedback to refine the Alloy model. VDS is designed to support short product release cycles. The AAEP separates domain-specific from domain-independent aspects so the VDS can be rapidly retargeted.**

I. Introduction

Cyber Physical Systems (CPS) are becoming increasingly complex, and are upgraded more frequently. Systems are composed of components, with vulnerabilities, some that are known publically, some that are known by the manufacturer, and some that are yet to be discovered. Just as a composite CPS can display emergent behaviors, i.e. behaviors that are not predicted solely by modeling the assembly of the constituent parts, it can also have **emergent vulnerabilities** leading to exploits that are unique to the aggregate system. These exploits could be catastrophic even though they result from a combination of relatively benign subsystem vulnerabilities. The composition of many components in a CPS leads to a combinatorial explosion of the behaviors to check. Given the short upgrade cycle and the rapidly changing list of vulnerabilities for each part, reaching a minimal level of due diligence when designing and building a CPS is a daunting task.

With automobiles becoming self-sufficient, even to the point of controlling the driving, the human becomes a passenger. This represents a ubiquitous, massively deployed, safety critical CPS in the center of our critical transportation infrastructure. Though this autonomy promises great improvements to public safety, it also poses potential threats to it. For example, the effect of a random car malfunction killing people, or a coordinated malfunction causing havoc in the transportation infrastructure.

This type of CPS is highly complex with a large set of sensors and information sources assisting the onboard command and control system to safely "drive" the vehicle on behalf of the human passengers. The manufacturers of today's commercial and defense systems struggle to prove that they work under all conditions. Industry has not even started looking at attacks.

In a non-cooperative environment the challenge is to prove the correctness of the CPS under all possible conditions. This task is extremely difficult based on the vast set of possible scenarios and the failure modes of the components involved. VDS offers a disciplined approach that identifies vulnerabilities in the system, selects scenarios that would expose those vulnerabilities, and generates controlled tests based on those scenarios. Though this approach still produces a very large set of tests, and has issues with how to compute robustness of the solutions, it is tractable with a massively parallel simulation and storage system.

VDS accelerates the detection of emergent exploits by identifying a manageable prioritized checklist of device-specific, feasible attack scenarios. These guide the discovery of automotive component vulnerabilities at a tempo supporting short product update cycles by:

- Automating ingestion of public and private vulnerabilities into a structured, semantically consistent format,
- Organizing vulnerabilities based on potential attack scenarios and transforming vulnerability information into standardized APs,
- Using an Hierarchical Task Network (HTN) planner to explore the APs and generating a large set of composite, device level sub-plans and attack scenarios to segment feasible and infeasible attacks against a known device,
- Using a satisfaction constraint model of the CPS, based on system constraints, to exhaustively evaluate device characteristics for potential exploit techniques, prune unlikely or infeasible attacks, and validate sub-plans to reduce the size of the set of composite vulnerabilities, and
- Handing off a reduced, prioritized list of weaknesses, based on the potential for damage, for execution in simulated and actual automotive devices, to validate weakness checklist guidance or provide constraint/success feedback to our planner and constraint satisfaction steps.

The possible permutations of vulnerabilities, physical system configurations and attack scenarios are numerous. This VDS method of automatically maintaining these ontological relationships is scalable, saves time, and reduces errors. Reducing the number of these permutations, using planning and feedback, yields a manageable set of vulnerabilities and attacks

to validate and check. The VDS results in earlier detection of potential exploits than is possible with manual methods, reducing cost and increasing safety.

The VDS is domain independent, but we currently develop and demonstrate in the context of the automotive domain. VDS analyzes emergent vulnerabilities of automotive component systems, including curve speed warning, traffic sign recognition, distance alert, and laser cruise control, on both an existing instrumented automobile and a simulation environment.

As shown in Figure 1, our approach consists of three parts: Knowledge Acquisition, Attack Plan Generation, and Attack Verification. Attack Verification has an existing implementation that has been successfully applied in the automotive domain. Set-up of the Attack Verification process is time-consuming and limits its effective application to rapidly evolving attack scenarios. VDS solves this problem by introducing extensive automation for attack scenario generation. We use the Attack Verification results to generate feedback that enables the system to incrementally refine the generated attack plans. The work described for Knowledge Acquisition and Plan Generation is currently at the proposal stage.

II. System Architecture

Figure 1 depicts the Knowledge Acquisition, Attack Plan Generator, and Attack Feedback and Generation architecture parts. These 1) generate a prioritized checklist of software and firmware, and classes of malicious functionality to rule out and 2) verify our attack/vulnerabilities hypothesis.

A. Knowledge Acquisition

To detect CPS threats, we consolidate data from several external data sources and restructure it into a format suitable for planning. Based on this team's study of MAISSI (Mediation, Alignment and Information Systems for Semantic Interoperability) [3], BAE Systems developed Dynamic Composition of Enterprise Services (DCES), an innovative approach to semantic alignment that handles the intricacies of real-world data.

Translation from source data into a semantically equivalent representation in a target ontology faces a number of problems. These include 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. MAISSI transforms each external data source record into a standardized AP record based on an Attack Surface Ontology (ASO) to normalize the data. We develop the ASO to accomplish two objectives: 1) translate vulnerabilities into potential APs and 2) capture attack-centered data that may include exploits, configurations, and ways to leverage legitimate device functionality.

A comprehensive AP 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. These are critical for developing a clear attack model that includes vulnerabilities and legitimate features that an attacker could misuse. This way, we enhance our attack surface perspective of a given device.

B. Attack-Centric Analysis and Context

The common bottom-up technology-centric perspective promoted by security researchers comes at the expense of understanding how a hacker discovers and exploits vulnerabilities. Hackers champion context, seeking to understand how a system functions in a top-down manner to reach a target. Each weakness grants the hacker leverage to gain greater control. Vulnerability repositories only organize and categorize system vulnerabilities rather than express how an individual vulnerability or feature may relate to, and even anticipate, an attack objective.

This VDS approach constructs AP systems to conduct an attack-centric analysis that examines how vulnerabilities and legitimate device services relate to attack objectives. A system represents a combination of services, configurations, functions and vulnerabilities that, when viewed from an attack perspective, combine into attack graphs depicting multistage cyber-attacks against systems [4]. Whereas Sheyner and Wing's research focuses on the layers within a complex network environment, we extend the attack graph definition to represent system services and functions as discrete attack objects, focusing on layers within a complex network environment. Each of those architectural objects represents potential APs that are accessible only from the next layer, with externally available services residing on the outer-most architecture layer.

The process of identifying and prioritizing system vulnerabilities involves reasoning over many series of exploits that can rapidly lead to an undesirably huge set of states. The VDS approach uses a predefined list of common attack patterns 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. 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

- Improve and direct search and test,
- Improve the goal generation process with spiked goals and plan fragments, and
- Refine structures with learned behavior.

This also facilitates parallel development and test. Figure 2 illustrates how the Semantic Alignment Mediator supports the VDS solution. In this example, to transform vulnerability data into attack-centric AP records, we combine the output from the Attack Goals analysis and an applicable National Vulnerability Database (NVD) [5] 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. This unification is achieved with declarative mappings from the Attack Goals repository and NVD schemas to the common ASO ontology. The result is a set of RDF statements in the AP knowledge

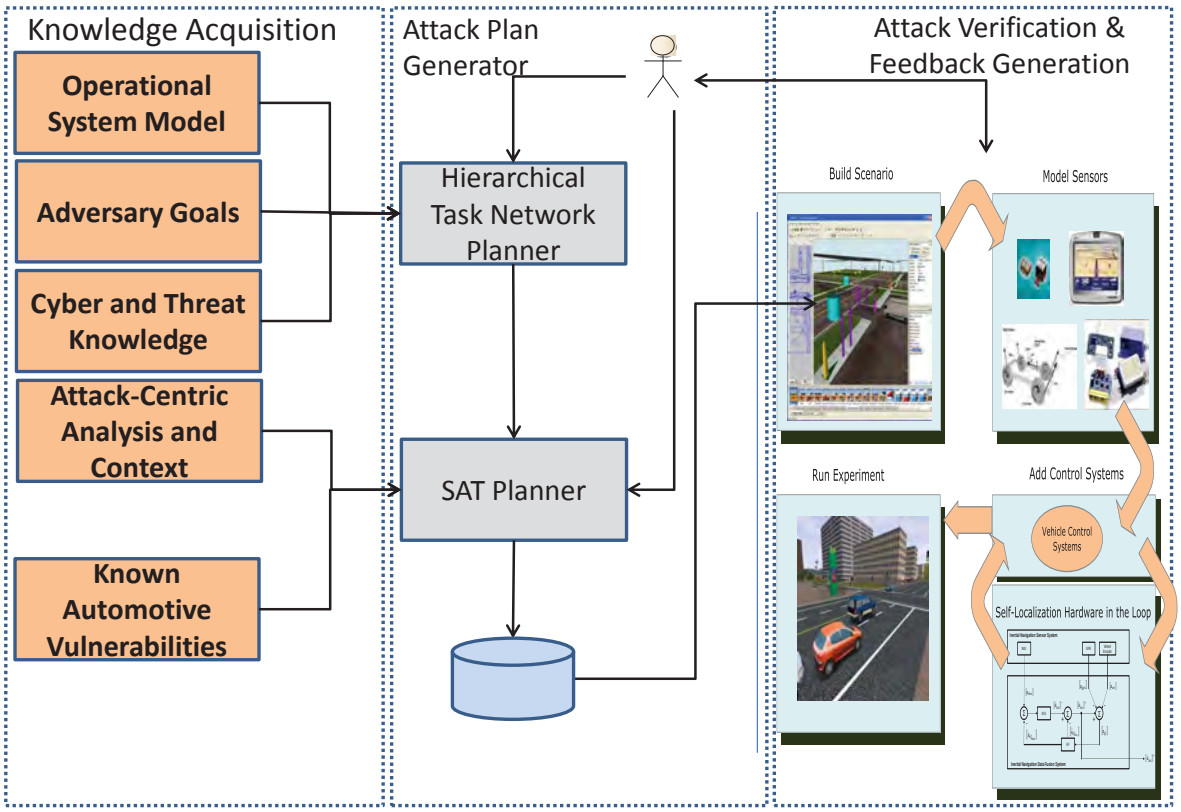

Fig. 1. System Overview

base that links a device with a particular vulnerability that can be analyzed to confirm malice. Common Attack Pattern Enumeration and Classification (CAPEC) [6] 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. These will be translated using the semantic alignment techniques discussed above into forms that the Attack Plan Generator (next section) can reason over.

C. Attack Plan Generation

VDS uses a hybrid planning mechanism and multiple sources of information to discover feasible attack plans. Alloy's [2] 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 uses 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 [7]), the device class architecture, and adversarial goals and strategies.

The planning engine supports checking for invalid preconditions 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. 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. The VDS 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 HTN planner and creates a minimally complete plan. An Activity Estimator uses the HTN's leaf nodes to create an initial Activity Graph, which embodies at least one chain of causally ordered qualitative states that connect the start state and a goal state.

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 the plan, or refuting the plan by demonstrating no instantiation is possible. Combining a heuristic search mechanism with Alloy enables reasoning from a diverse set of information and compensates for weaknesses of the individual planners when used in isolation. Search prioritizes high-risk candidates based on a

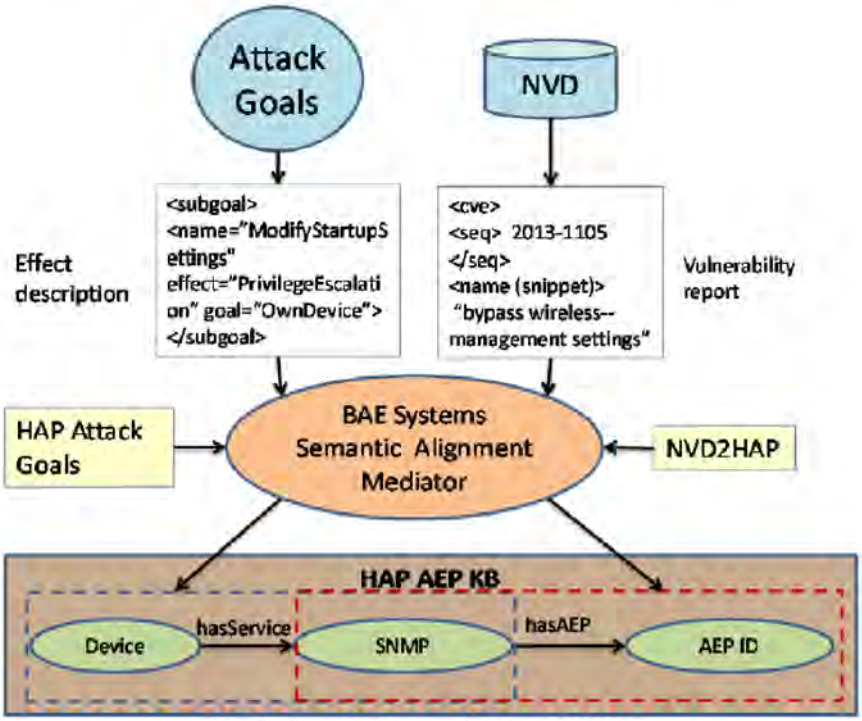

Fig. 2. Semantic Alignment Mediator (MAISSI)

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 automotive component being assessed, made specific through assumptions in the heuristic search to exhaustively check for existence of a sequence of AP-level actions that can achieve adversary goals. 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 $Sys \wedge \neg P$).

Given a set of feasible, concrete plans generated by Alloy, combined with the calculated likelihoods of different configurations, the VDS produces a checklist that is ordered by risk and likelihood. For the achievable attacker goals that pose the greatest risk, VDS groups together attack plans achieving similar goals, and identifies required configuration settings.

The primary value of the Knowledge Acquisition and Attack Plan Steps is to generate a checklist as an input to focus more detailed analysis. This checklist is prioritized and actionable to make best use of limited analyst resources. Approaches that perform a post-processing prioritization step would be hopelessly inefficient. The checklist VDS generates is in the order of likelihood of attack success.

D. Attack Verification and Feedback Generation

As a prerequisite, a large set of scenarios (based on real-world driving data) are abstracted into a virtual world model. This large data set is then tagged with descriptors for each scenario.

Attack Verification and Feedback (AVF) generation uses the vulnerabilities identified through the feasibility and risk prioritized attack plan checklist, then selects scenarios that would expose those vulnerabilities, and generate controlled tests based on those scenarios.

Testing all possible conditions of a CPS in a non-cooperative environment is difficult due to the vast set of possible scenarios and failure modes. The attack plan checklist selects scenarios that would expose those vulnerabilities and generate controlled tests. By combining this technology with massively parallel computation, storage system, and simulation technology, VDS generates parameterized scenarios based on real-world data, and run them in parallel.

Driving tests, and simulation systems that run a multiple scenario closed loop simulation with a System Under Test (SUT) are executed. The test generator selects test cases out of the collection that likely expose predicted vulnerabilities. These cases are then run using a driving and sensor simulator (PreScan by TASS [8]), a model of the vehicle (CPS) under test (Matlab/Simulink) and a vehicle dynamics model

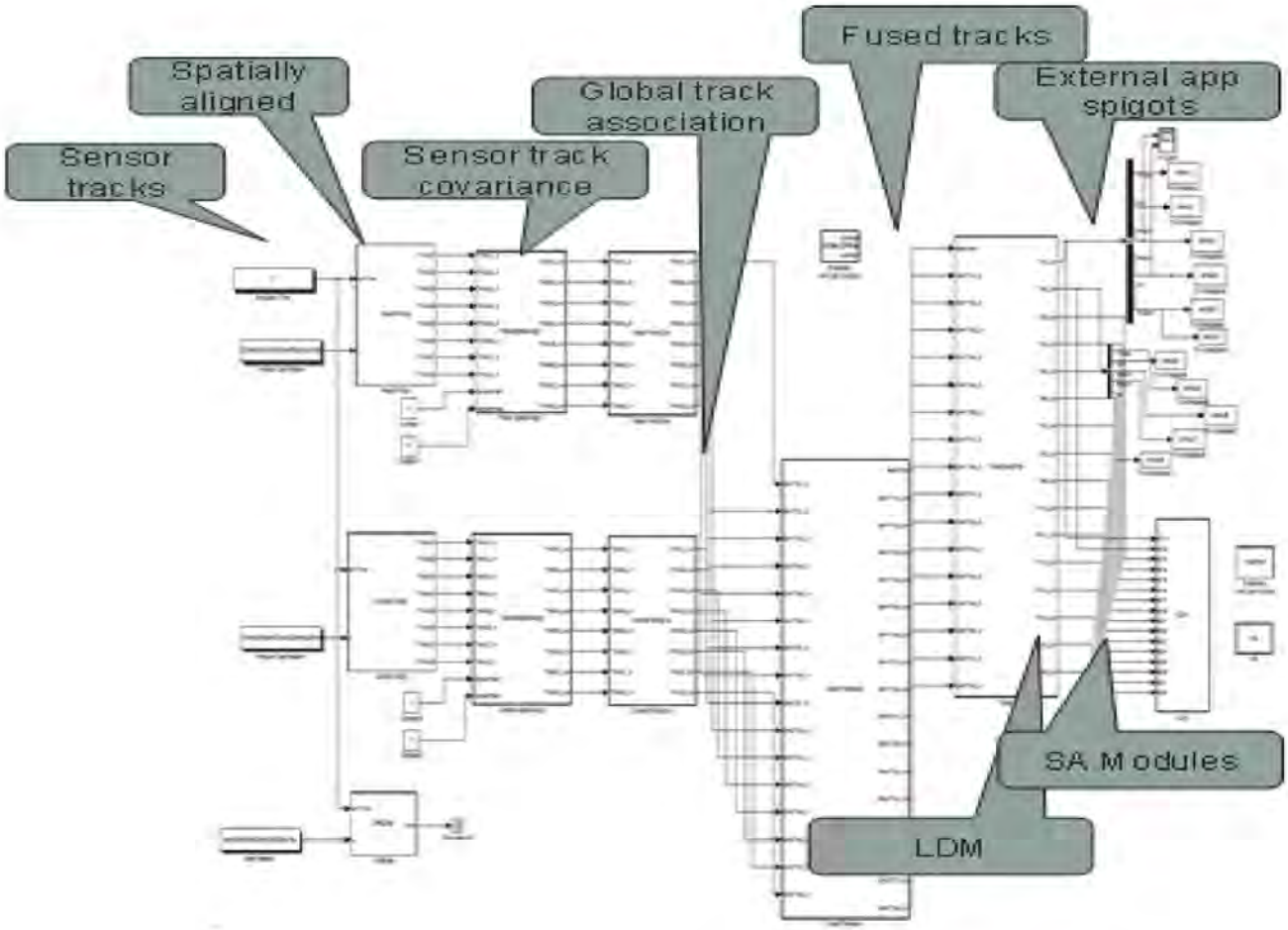

Fig. 3. Executable Simulink Specification is System Under Test that runs on simulation and automotive testbeds

in a closed loop. Simulation results (pass, fail, near miss, emergency, etc.) are fed back to the test generator where modified test scenarios are generated creating the outer loop. Well-defined interfaces of separately pluggable independently developed modules are used to enable scaling and modifying of applications independent of real or simulated sensors. Figure 3 shows Simulink correspondence to existing automotive software that runs the same on computer and test-car test benches. Both positive and negative test bench results are returned to the Attack Plan Generator, with explanation, to improve the HTN and SAT planner process.

In our second use case the System Under Test (SUT) is exposed to an external attack. In this case the proposed system generates attacks scenarios on the systems.

In these case the "real-world" scenarios are taken from the recorded sample set, but then the "test" scenarios are modified by the attackers approach in the simulator (for example target appear in different locations for the radar sensor and camera sensor, or a GPS drift is imposed). The test generator then samples the parameter space of this scenario based on results, and randomly varies the scenario (1000's of different variations) to try to expose vulnerabilities in the system. The found vulnerabilities are then fed back to the system and/or user for further analysis.

The parameter space for these systems is so vast that a brute force method for securing and proving correctness is not feasible. With this approach the search space is reduced to manageable dimensions while maintaining "full" test coverage.

III. 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 [3] 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.

REFERENCES

[1] J. Smith and M. Figueroa, "Reduced realistic attack plan surface for identification of prioritized attack goals," in *2013 IEEE Homeland Security Conference*, Nov. 2013.

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

[3] BAE Systems, "Mediation, alignment, and information services for semantic interoperability (MAISSI): A trade study," Air Force Research Laboratories, Tech. Rep. AFRL-IF-RS-TR-2007-147, June 2007.

[4] O. Sheyner and J. Wing, "Tools for generating and analyzing attack graphs," in *2nd Intl. Symposium on Formal Methods for Components and Objects*, ser. LNCS, no. 3188. Springer Verlag, 2004.

[5] NIST, "National vulnerability database." [Online]. Available: http://nvd.nist.gov

[6] The MITRE Corporation, "Common attack pattern enumeration and configuration." [Online]. Available: http://capec.mitre.org

[7] ——. Common vulnerabilities and exposures. [Online]. Available: http://cve.mitre.org

[8] (2014) PreSCAN product page. [Online]. Available: https://www.tassinternational.com/prescan