T H E U N I V E R S I T Y O F T U L S A

THE GRADUATE SCHOOL

HYBRID ATTACK GRAPHS FOR MODELING CYBER PHYSICAL SYSTEMS SECURITY

by
George Robert Louthan IV

A thesis submitted in partial fulfillment of
the requirements for the degree of Master of Science
in the Discipline of Computer Science

The Graduate School
The University of Tulsa

2011

T H E U N I V E R S I T Y O F T U L S A

THE GRADUATE SCHOOL

HYBRID ATTACK GRAPHS FOR MODELING CYBER PHYSICAL SYSTEMS SECURITY

by
George Robert Louthan IV

A THESIS

APPROVED FOR THE DISCIPLINE OF

COMPUTER SCIENCE

By Thesis Committee

_____, Chairperson
John C. Hale

_____
Mauricio Papa?

_____
Peter Hawrylak?

ABSTRACT

George Robert Louthan IV (Master of Science in Computer Science)

Hybrid Attack Graphs For Modeling Cyber Physical Systems Security

Directed by John C. Hale

18 pp., Chapter 1: Conclusions

(75 words)

As computer systems' interactions with the physical world become more pervasive, largely in safety critical domains, the need for tools to model and study the security of these so-called cyber physical systems is growing. This thesis presents extensions to the attack graph modeling framework to permit the modeling of continuous, in addition to discrete, system elements and their interactions, to provide a comprehensive formal modeling framework for describing cyber physical systems and their security properties.

ACKNOWLEDGEMENTS

Thanks to Evan Mackay and my family for their constant support.

More acknowledgments go here.

This material is based on research sponsored by DARPA under agreement number FA8750-09-1-0208. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, or DARPA or the U.S. Government.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

As computer systems become pervasive across a variety of domains, not only are their interactions with people becoming more frequent; computer systems are also increasingly interacting with the physical world and with each other.

Systems that include both continuous and discrete components are termed *hybrid systems*. When linked together with a significant network component, these systems are sometimes called *cyber physical systems*, which have been targeted as a key area of research. Such systems are becoming pervasive in safety-critical domains such as medical, critical infrastructure, automotive, and others. This thesis is concerned with modeling the security of these systems and their interactions with each other and the physical world.

1.2 Modeling Frameworks

An excellent argument that touches on the need for new research directions in modeling cyber physical systems is due to Lee in a 2006 position paper in the National Science Foundation Workshop on Cyber-Physical Systems, a prelude to the NSF's research initiative on cyber physical systems:

> Cyber-Physical Systems (CPS) are integrations of computation with physical processes. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect compu- tations and vice versa. In the physical world,

> the passage of time is inexorable and concurrency is intrinsic. Neither of these properties is present in today's computing and networking abstractions. [13]

Existing frameworks for modeling and analysis of computer networks are inappropriate for use in these systems because of their inability to capture the continuous domain; they also lack a robust, let alone "inexorable" notion of time. Likewise, existing methods for studying hybrid systems fall short when it comes to modeling the sometimes complex networks that are hallmarks of cyber physical systems.

1.3 Scope

This thesis presents an extension of the attack graph modeling framework, typically used for studying network security, into the continuous domain to enable it to be used for studying cyber physical systems. The goal is to combine the best of both worlds: hybrid systems modeling frameworks, particularly hybrid automata, which best describe systems in relative isolation; and computer network security modeling frameworks, particularly attack graphs, which excel at capturing the complex interrelationships and interdependencies between assets and attacks.

The remainder of this thesis is structured as follows. Chapter 2 provides background in hybrid systems and their modeling methods, introduces past work in attack graphs, and presents a set of case studies in both the hybrid and discrete domains to be used throughout this work. Chapter 3 introduces in detail the basic attack graph framework to be used as the basis for the hybrid extensions. Chapter 4 introduces the extensions themselves. Chapter 5 delivers some results from this modeling methodology, and Chapter 6 draws conclusions and suggests further work.

CHAPTER 2

BACKGROUND

2.1 Cyber Physical Systems

2.1.1 Hybrid Systems

A system with both continuous (frequently physical) components and discrete (frequently digital) components is said to be a *hybrid system*, named for its characteristic blending of the two domains. Examples of hybrid computer systems abound in industrial controls, for example, although hybrid systems may also be fully physical (e.g., a bouncing ball experiences continuous behavior when rising and falling and discrete behavior when colliding with a surface).

The term hybrid system is an older one that was coined as researchers began to study the newly pervasive reactive systems that arose as programmed control of the physical world became widespread [2]. For several reasons it does not suffice to describe precisely the types of systems with which this work is concerned: a subset of hybrid systems that incorporate a significant computer and networking component.

Nevertheless, the modeling of hybrid systems is well studied and provides a sufficient body of relevent knowledge from which to draw to warrant its inclusion. This chapter includes background on a particularly relevent modeling framework for hybrid systems called the hybrid automaton, which is used in this thesis as the standard benchmark against which to compare hybrid modeling techniques.

2.1.2 Cyber Physical Systems

DefinitionA newer, better term for the systems investigated in this thesis is *cyber physical systems*. Put simply, a cyber physical system is a networked hybrid system: a networked computer system that is tightly coupled to the physical world.

ChallengesAccording to the 2008 Report of the Cyber-Physical Systems Summit, "The principal barrier to developing CPS is the lack of a theory that comprehends cyber and physical resources in a single unified framework." [1]

The summit further identified as part of the scientific and technological foundations of cyber physical systems both new modeling frameworks that "explicitly address new observables" and studies of privacy, trust, and security including "theories of cyber-physical inter-dependence" [1], a major theme of this work.

Crenshaw and Beyer enumerated four principal challenges in cyber physical systems testing that are equally apt for security: their concentration in safety critical domains, their frequent integration of third-party or otherwise unrelated systems, their dependence upon unreliable data collection, and their pervasiveness [5].

2.1.3 Hybrid Automata

DefinitionA valuable formalism for modeling hybrid systems in isolation and with limited composition is the hybrid automaton of Alur, et al. [2]. This section introduces the version of the formalism described in 1996 by Henzinger [8], to which a reader interested in more than a superficial understanding is referred.

Formally, a hybrid automaton H is made up of a set of real-valued state variables, their first derivatives, a set of operational modes and switches between the modes, and predicates attached to those modes and switches describing the operation of the system in those modes and the discrete transitions between them. One can think of a hybrid automaton as a pairing of a finite state machine whose states (called modes) and transitions (called switches) denote the discrete-domain

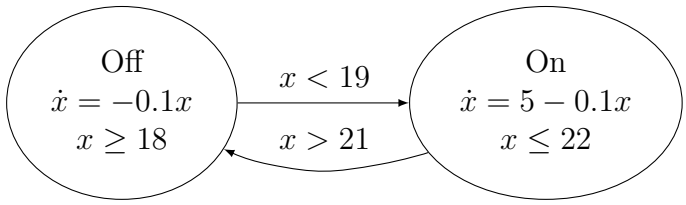

Figure 2.1: Thermostat hybrid automaton [8]

behavior of the hybrid system, with a set of differential equations attached to each mode, which govern its continuous-domain behavior. Switches may also be labeled in order to permit synchronization across composed hybrid automata.

Modes may be decorated with invariant conditions (which state whether the system is allowed to be in that mode), flow conditions (which state how the continuous domain state variables are permitted to evolve while in that mode), and initial conditions (which state under which, if any, conditions the automaton may begin its operation with that mode). Switches are decorated with jump conditions, which serve as guards on the switch determining both (1) when the switch is allowed to be taken, and (2) the discrete changes in state variables due to that switch's activation.

A simple example of a hybrid automaton is given in Fig. 2.1, which models a simple heater thermostat. The nodes in the automaton represent its operating modes, and the edges represent switches. In the "Off" mode, the temperature (given by x) must be greater than or equal to 18, and its first derivative with respect to time (denoted $\dot{x}$) is $-0.1x$, which represents a cooling of the environment. When the temperature is strictly less than 19, the switch from off to on is available (but not mandatory until the off mode's invariant condition $x \geq 18$ ceases to be satisfied. The switch from on to off behaves similarly.

The hybrid automaton model is sufficiently rich to capture many hybrid systems.

ShortcomingsThere are some problems with the hybrid automaton model. A hybrid automaton is not guaranteed to have a valid execution, and computing whether it does or not is non-trivial [15]. Model checking has been developed for only some subclasses of automata [9] [7], and many desirable properties of them are undecidable [10].

However, there are even more nagging problems when considering hybrid automata or their variants for the study of cyber physical systems. One of the hallmarks of cyber physical systems is a distributed and highly networked nature. While they provide a natural model for the discrete-continuous boundary, hybrid automata have only a rudimentary notion of communication, no clear means for specifying message passing, and when used in large topologies have significant scaling problems, both computationally and cognitively.

AlternativesSome attempts have been made to solve the problem of the hybrid automaton's unsatisfactory capability for modeling networks and communication. Particularly, the designation of shared actions and shared variables as "input" or "output" is a popular tactic, used in the powerful hybrid I/O automaton [17] [16], its descendent the timed I/O automaton [12], and also in the PHAVer model checker [7].

The work of this thesis is also something of an outgrowth from an instance of this strategy in which prototypical "hybrid link automata" were developed to model explicit communication channels. An example of the cognitive scalability issues inherent with this design is given in Fig. 2.2. This strategy may have a place in modeling some systems but falls short of the goal of modeling complex, interdependent networks of hybrid systems with more conventional computer networks.

2.2 Attack Graphs

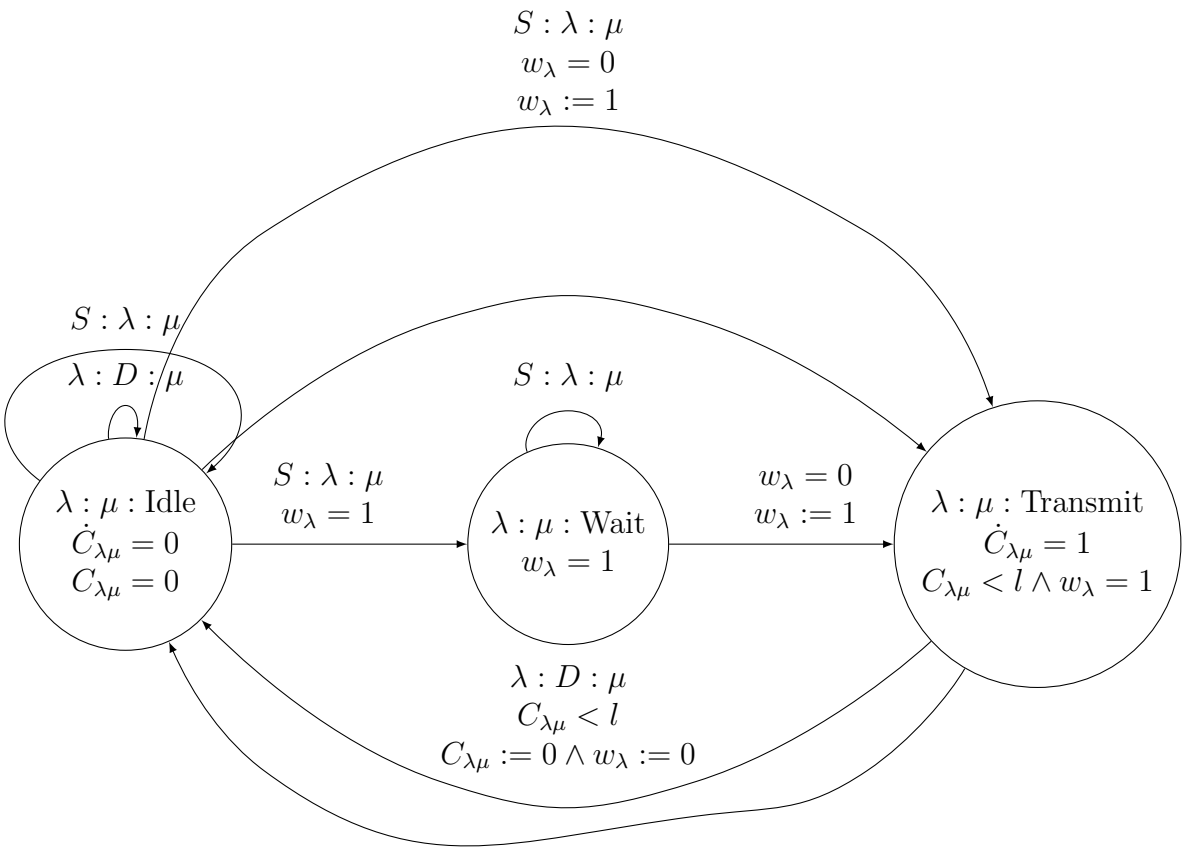

Figure 2.2: An example of a considered "hybrid link automaton" prototype modeling a link on which messages may be dropped, injected, or delayed, and on which mutual exclusion of messages is enforced.

2.2.1 Introduction

An attack graph is one of several related formalisms that utilize graph theory to model the state space of computer systems attacks. Perhaps they are best introduced when presented as an alternative to a similar model called an attack tree.

2.2.2 Attack Trees

An attack tree is a goal-oriented tree model of an abuse of a system [19]. The root of the tree represents the attacker's goal, and the children of any given node represent the prerequisite activities required to reach that node. For example,

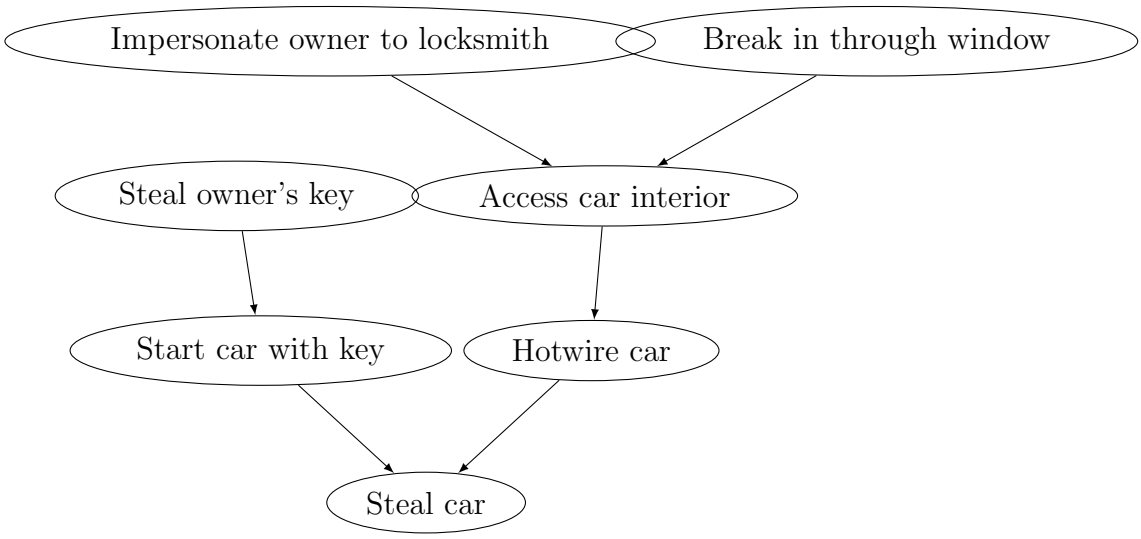

Figure 2.3: Simple car theft attack tree

consider the goal of stealing a car, which is modeled in a simple attack tree in Fig. 2.3.

The attacker must start the car and drive away; this could be accomplished either by breaking in and hotwiring the car, or by stealing the owner's key and using it to subsequently steal the car. The root of the tree represents the final goal of the theft, with prerequisite goals flowing upward from the leaf nodes.

There are a few features of this modeling method to note. It is goal oriented, meaning that the consequences of the attack are known, and the goal is to ennumerate and and analyze the means by which those consequences could be reached. It is, as an attack model, agnostic to the underlying system model which makes it difficult to generate automatically. Finally, and perhaps most significantly, it captures the ways in which an attacker's actions interact and depend upon each other.

This threat-centric model is not necessarily the most useful for system stakeholders. It requires, in a sense, that one work backward from the attack to the system state necessary to realize the attack. If, instead, an analyst desires to work from a system characterization and explore the attack space permitted by that

system characterization, the attack tree framework must be in some sense turned upside down. Attack graphs do exactly that.

2.2.3 Attack Graphs

IntroductionIn contrast to attack trees, attack graphs permit a topology-aware exploratory analysis of the state space of a system. It is a graph theoretic model in which vertices represent individual system states, and edges represent state transitions caused by an adversary. The concept as introduced in 1998 included notions of generalized attack patterns to be bound to state transitions; network elements and their individual configurations; network topology (three characteristics common to all current attack graph iterations); a notion of the attacker's capabilities, and edge weights representing likelihood [18]. A similar structure called a privilege graph was introduced in 1994 [6].

Most approaches to attack graph modeling represent exploits (attack patterns) as using preconditions and postconditions [14] since this was suggested in about 2000 [21]. Exploits are chained together by matching preconditions in a state node's underlying system model and applying their postconditions to generate a successor state.

Model TypesThe modeling substrates of attack graphs can be broadly separated into two schools of thought, separated by the philosophy that guides the representation of the underlying network model over which network states and transitions are computer.

A specification of an underlying network model may be done with only very loose restrictions, allowing arbitrary keywords as named qualities and topoligies of network objects. This thesis employs this method. It is also favored in the work of George Mason University [3] [23]. It has the advantage of permitting more

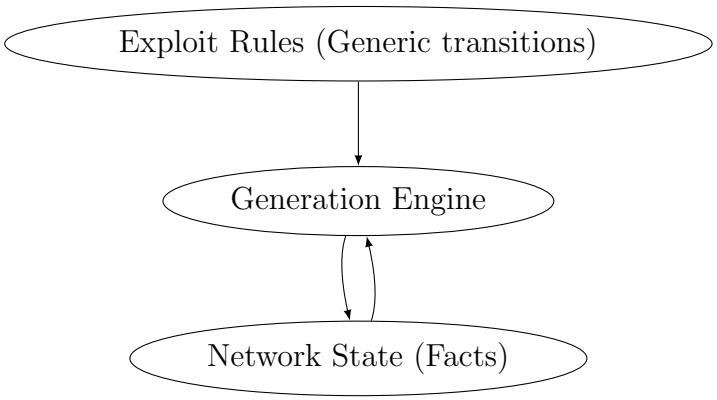

Figure 2.4: Attack graph generation process

straightforward adaptation into the continuous domain, which is the reason it is favored by this work.

An alternate specification method is much more restricted, confining the modeler to certain sets of terms imposing explicit computer networking comcepts onto the model [21]. This permits generation and analysis to take advantage of networking concepts to perform a more nuanced analysis of a network state, including reachability analysis to determine whether a given topology permits communication between two hosts [11]. This approach is favored, for example, in the work of MIT Lincoln Laboratory and the University of California, Davis.

GenerationMethods of attack graph generation, the process of chaining exploits to enumerate the attack space [4] [18] [20], share a common general architecture among the modern methods that use preconditions and postconditions in exploit definitions, pictured in Fig. 2.4. The attack graph generation process combines network state and exploit patterns as input, applying exploit postconditions back onto the network state to generate its output of successor states.

Research DirectionsResearch in attack graphs is spread throughout a variety of pathways. Many of these include evaluating a network's security [3], specification of formal languages to represent attack graphs [21], intrusion detection system

integration [22], automatic generation of security recommendations [23], and reachability analysis between hosts in a single network state [11]. For a thorough literature review up to 2005 and more detailed discussion of popular research directions, refer to the work of Lippmann and Ingols [14].

2.3 Case Studies

CHAPTER 3

ATTACK GRAPHS

3.1 Definition

3.2 Working Lexicon

3.3 State Predicates

3.4 State Aggregation

CHAPTER 4

HYBRID EXTENSIONS

4.1 Introduction

4.2 Definition of New Syntax

4.3 Time

4.4 Time State Aggregation

CHAPTER 5

RESULTS

CHAPTER 6

CONCLUSION AND FUTURE WORK

BIBLIOGRAPHY

[1] Report: Cyber-physical systems summit. Technical report, National Science Foundation, 2008.

[2] R. Alur, C. Courcoubetis, T. Henzinger, and P. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. *Hybrid systems*, pages 209–229, 1993.

[3] P. Ammann, D. Wijesekera, and S. Kaushik. Scalable, graph-based network vulnerability analysis. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 217–224. ACM, 2002.

[4] C. Campbell, J. Dawkins, B. Pollet, K. Fitch, J. Hale, and M. Papa. On Modeling Computer Networks for Vulnerability Analysis. *DBSec*, pages 233–244, 2002.

[5] T.L. Crenshaw and S. Beyer. UPBOT: a testbed for cyber-physical systems. In *Proceedings of the 3rd international conference on Cyber security experimentation and test*, pages 1–8. USENIX Association, 2010.

[6] M. Dacier and Y. Deswarte. Privilege graph: an extension to the typed access matrix model. *Computer Security ESORICS 94*, pages 319–334, 1994.

[7] G. Frehse. PHAVer: Algorithmic verification of hybrid systems past HyTech. *Hybrid Systems: Computation and Control*, pages 258–273, 2005.

[8] T.A. Henzinger. The theory of hybrid automata. In *Logic in Computer Science, 1996. LICS'96. Proceedings., Eleventh Annual IEEE Symposium on*, pages 278–292. IEEE, 1996.

[9] T.A. Henzinger, P.H. Ho, and H. Wong-Toi. HyTech: A model checker for hybrid systems. *International Journal on Software Tools for Technology Transfer (STTT)*, 1(1):110–122, 1997.

[10] T.A. Henzinger, P.W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata? *Journal of Computer and System Sciences*, 57(1):94–124, 1998.

[11] K. Ingols, M. Chu, R. Lippmann, S. Webster, and S. Boyer. Modeling modern network attacks and countermeasures using attack graphs. In *2009 Annual Computer Security Applications Conference*, pages 117–126. IEEE, 2009.

[12] D.K. Kaynar, N. Lynch, R. Segala, and F. Vaandrager. The theory of timed I/O automata. *Synthesis Lectures on Distributed Computing Theory*, 1(1):1–137, 2010.

[13] E.A. Lee. Cyber-physical systems-are computing foundations adequate. In *Position Paper for NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap*. Citeseer, 2006.

[14] R.P. Lippmann, K.W. Ingols, and MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB. *An annotated review of past papers on attack graphs*. Massachusetts Institute of Technology, Lincoln Laboratory, 2005.

[15] J. Lygeros, K.H. Johansson, S. Sastry, and M. Egerstedt. On the existence of executions of hybrid automata. In *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, volume 3, pages 2249–2254. IEEE, 1999.

[16] N. Lynch, R. Segala, and F. Vaandrager. Hybrid I/O automata revisited. *Hybrid Systems: Computation and Control*, pages 403–417, 2001.

[17] N. Lynch, R. Segala, F. Vaandrager, and H. Weinberg. Hybrid I/O automata. *Hybrid Systems III*, pages 496–510, 1996.

[18] C. Phillips and L.P. Swiler. A graph-based system for network-vulnerability analysis. In *Proceedings of the 1998 workshop on New security paradigms*, pages 71–79. ACM, 1998.

[19] B. Schneier. Modeling security threats. *Dr. Dobb's journal*, 24(12), 1999.

[20] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J.M. Wing. Automated generation and analysis of attack graphs. 2002.

[21] S.J. Templeton and K. Levitt. A requires/provides model for computer attacks. In *Proceedings of the 2000 workshop on New security paradigms*, pages 31–38. ACM, 2001.

[22] T. Tidwell, R. Larson, K. Fitch, and J. Hale. Modeling internet attacks. In *Proceedings of the 2001 IEEE Workshop on Information Assurance and security*, volume 59, 2001.

[23] L. Wang, S. Noel, and S. Jajodia. Minimum-cost network hardening using attack graphs. *Computer Communications*, 29(18):3812–3824, 2006.