Security Modeling in the COTS Environment

Tom Markham, tom_markham@securecomputing.com
Dwight Colby, dwight_colby@securecomputing.com
Secure Computing Corporation, 2675 Long Lake Road, Roseville, MN 55113-2536

Mary Denz, denzm@rl.af.mil
Air Force Research Laboratory, AFRL/IFGB, 525 Brooks road, Rome, NY 13413-4505

Abstract

This paper is intended to stimulate discussion about the future of security modeling. It asserts that traditional top down security modeling is nonexistent today. However, security modeling tools that can address COTS based command and control systems are needed. The paper introduces several modeling paradigms, introduces requirements, then ends with a strawman intended to stimulate discussion among security researchers.

1. Security Modeling, Then and Now

Top down modeling based upon formal security policy models generated by wishful security researchers have gone the way of the punch card and the 8086. The 1980s mentality could be described as a security factory. We had the notion that we could specify a security model then build a system to implement the model from the design all the way down to the binary code. Projects such as SCOMP, PSOS, and LOCK actually developed formal security models.

The end of the cold war brought an end to much of the formal security policy work. Today the challenge is to build systems providing confidentiality, availability, and integrity from low assurance COTS products for which we don't write specifications, designs, or develop code. The security factory concept is dead.

Building a perfect security product to factory-like specifications is not practical because we do not control COTS vendor quality and the rules are constantly changing. The security factory has been replaced by a **security game**. It is a game because we no longer engineer security the way a civil engineer designs a bridge. Figure 1 compares modeling with games to show how the modeling environment has changed. COTS modeling is less strategic and more covert than the security factory because vendors determine features and bugs. They also restrict access to designs and source code.

1999 New Security Paradigm Workshop 9/99 Ontario, Canada

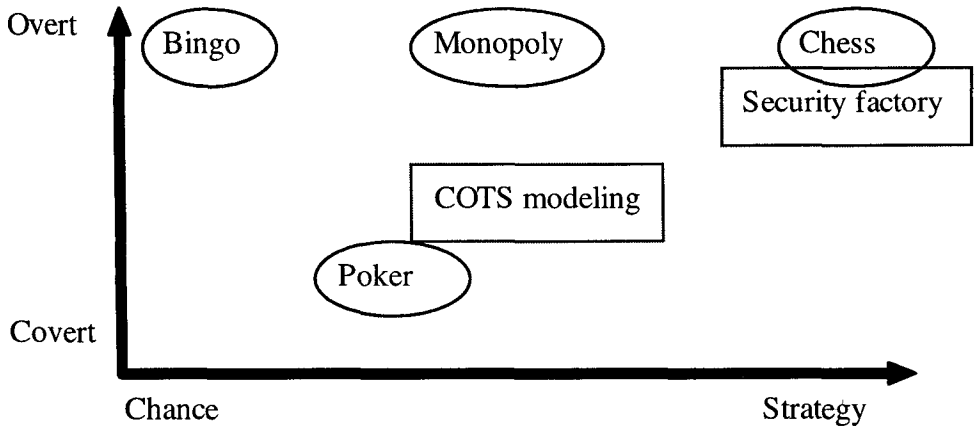

Figure 1 Security modeling compared with games

We are not working against a fixed, written set of criteria. We are working against a range of adversaries that are continually learning and acquiring new tools. The COTS products are continually being "upgraded" and new vulnerabilities are introduced with each upgrade. Thus, the problem cannot be addressed with the classical engineering approach of requirements, design, and development. The game is to figure out how to patch or hide the vulnerability before the adversary exploits it. The DARPA/AFRL Information Assurance program focus is on the protection of Command and Control computer-based information and systems, better known as "Cyber defense".

Given that we don't design secure systems from the ground up, do we need a security model at all? *Yes!* A good model abstracts from the raw data the facts that are fundamental and controlling. It establishes their interrelationship in ways that were not understood before (or were too complex to be handled within the mind), and generates predictions of surprising new facts. The model also acts as a vehicle, a common language, for communicating about security.

Electrical engineers know that if they cannot make a circuit work on paper or in a Computer Aided Design tool, then it will not work in the real world. The real world contains a wide range of problems that are not visible in most models. The same holds true for security engineering. If we do not have a model of how the system security is supposed to work, the chances of the system being secure in the real world are dismal. We must have a way to represent security mechanisms and the results we expect when operated in a specific environment.

The growing complexity of networks implies that without adequate tools the complexity of our systems will overtake our ability to comprehend them. In the *old days* there was TCP/IP, SMTP, FTP and Telnet. Today we have HTTP, IIOP, SSL, ISAKMP, IPSEC, H.323, DNS, and about two thousand RFCs. The number of transistors on a chip and the number of hosts on the Internet are both measured in millions. All of the trends imply increasing bandwidth, increased mobility, more applications, and more sharing of data. The complexity of securing all this is growing and will continue to grow at an exponential rate.

We must ask ourselves, "How can we win at the security game?"
Recent history provides an example of how American information technology defeated the Soviet challenge. The "challenge" was that of chess champion Garry Kasparov. The dominant information technology was IBM's Deep Blue. Deep Blue contained a model of chess and was capable of looking at 200 million moves/second. It was able to quickly look ahead at alternative courses of action and determine which move provided the optimal way to achieve the long term objective. IBM's strategy was actually very simple.

1. Develop a model of the game.
2. Look ahead, using the model, to develop the optimal course of action.
3. Observe your opponents move.
4. Loop back to step 2 until checkmate is achieved.

The key to winning the security game is to move from covert (unknown) to overt (known) and from chance (no model) to strategy (A well understood model). Intrusion detection and network discovery tools move us toward the overt. The development of functional models of COTS components is also necessary. The remainder of this paper is focused on developing a model of how systems are attacked and defended so that we can create strategy.

DARPA Information Assurance

2. What Should be the Paradigm for Today's Security Model?

The security factory models were primarily, but not exclusively, focused on confidentiality. The increased reliance on our networked systems demands that models also include integrity and availability. What type of model is best for illuminating vulnerabilities and keeping up with the stream of changes occurring in our systems?

The current environment forces us to build secure systems out of insecure components. We still have high level security requirements but the low level components such as protocols, operating systems, and applications are dealt to us by COTS developers. This implies that our security model must have a notion of top down, bottom up, and meet in the middle to create the best fit as shown in Figure 2.

Ideally the top down requirements and bottom up capability would meet perfectly in the middle and all of the top down security requirements would be satisfied. In reality, there may be gaps and the model must be capable of providing the information necessary to determine the security policy actually implemented.

One benefit of the instrumented model approach is that it allows the creation of new security mechanisms within the model relatively cheaply.

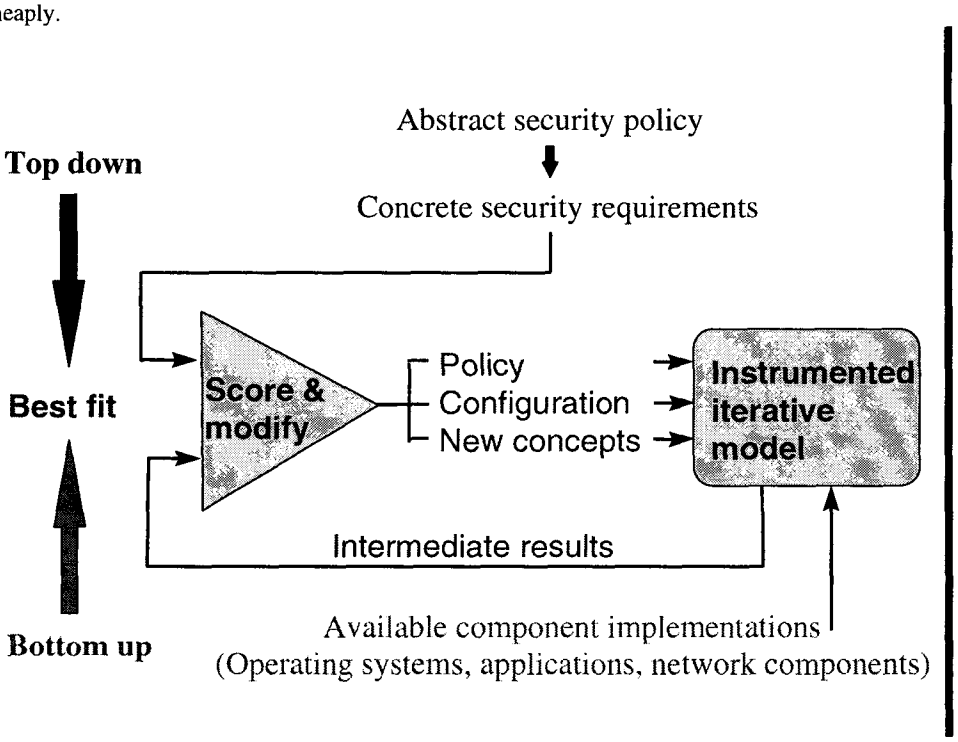

Figure 2. The "meet in the middle to create the best fit" approach.

The top down security factory model is no longer valid. This forces us to ask, "What modeling paradigms match today's cyber defense environment?" One approach is to look for models that have similar characteristics to those of the security systems/networks to be modeled. These provide insight in the characteristics of a good security model. Several modeling paradigms are discussed below.

- **Games.** Chess is an obvious example because it involves strategy and thinking several moves ahead of your opponent. However, chess is too overt for cyber defense. In cyberspace, you may not know who your opponent is, where they are, and what moves they are capable of making. Likewise, your opponent may not know exactly what system to take down in order to win. Stratego contains some of the elements of cyberwarfare because in Stratego your opponent does not know where your flag (your critical piece) is and you do not know the strength of an attacker until they attack. Chess lacks the element of chance but security is a game that includes chance. Unfortunately, configuration errors and some operating system crashes occur unpredictably. What are the random/chance elements that must be part of the model? Generally too much chance and probability turns the model to mush. As a result you cannot say anything useful and concrete about the security of a system if the model is overloaded with probability based guesses as opposed to clear decisions based upon specific, known configuration information. Still, it may be practical to assign probabilities to human behavior elements in the model, such as users' choice of passwords.

- **Biology.** Biological models are similar to security modeling in several respects. First, the biological world and COTS product worlds are not built to our specifications. We must perform experiments and build models derived from implementations, not from specifications. Informal security experiments have been performed on implementations of COTS products and the results are available in the CERT vulnerability databases. Second, both biology and networks involve many parameters that interact in subtle and complex ways. For example, the interaction of web browsers, Excel, and the operating system (the Russian New Year exploitation of Jan 1999) was not obvious from examination of any of the individual components. Finally, both biology and networked systems involve an element of chance. You don't know when an unwitting insider will import a virus or when your machine will crash.

- **Electrical Engineering Computer Aided Design (CAD).** CAD tools are ideal for handling a large number of elements and applying relatively simple rules to the elements. Modern microprocessors would not be possible without modeling every gate and its interaction with every gate connected to it. Techniques such as "design rule checking" and component libraries have increased productivity and reduced flaws.

One aspect that is not adequately represented by any of the models above is the rapidly evolving nature of security. Imagine playing chess if the capabilities of the game pieces changed every game. The strategy you developed for yesterday's game may not make sense for today's game. The top down models of the past implied that security researchers were working against a static set of evaluation criteria. Today, we are actually working in a dynamic environment that includes tactical and strategic events. Figure 3. Tactical and strategic feedback loops, illustrates the dynamics of today's situation.

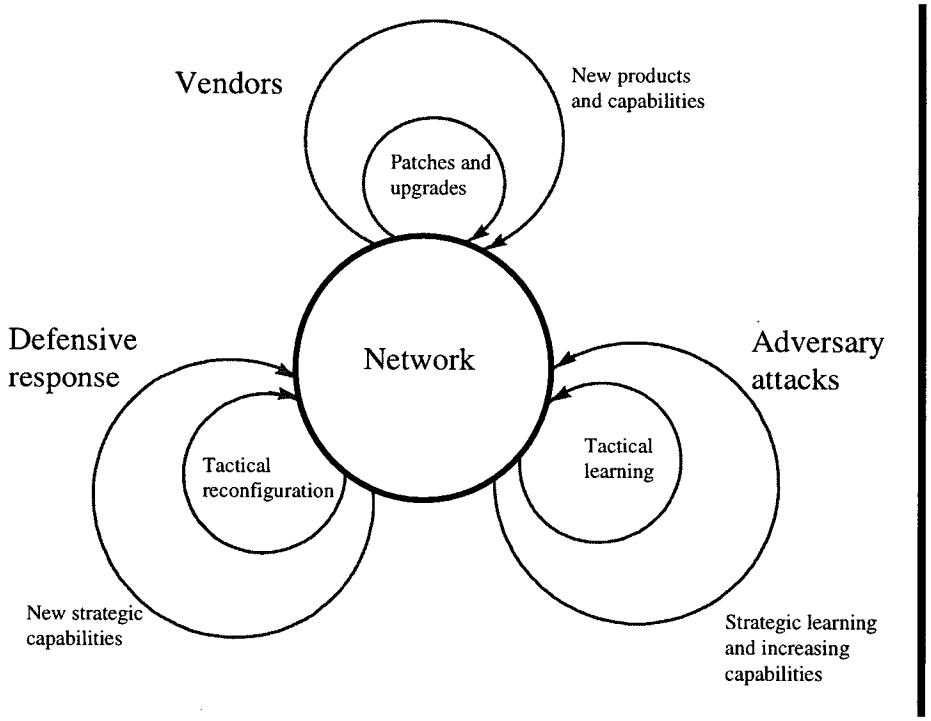

Figure 3. Tactical and strategic feedback loops

Vendors of network components (green) produce tactical patches and upgrades (service packs) to address near term vulnerabilities. They also develop new products (e.g., Java, PKI) and capabilities in a strategic sense. Our adversaries (red) probe and learn about our systems daily. Attackers are continually developing more sophisticated attack tools. On a strategic level they develop doctrine based upon their high level view of our defensive capability. The network defenders (blue) perform tactical reconfiguration (e.g., block an IP address) to respond to today's problems. We must develop new strategies and capabilities by looking out into the future. The key to effective security modeling, like the key to winning at chess, is to have a model that allows you to evaluate and select the best course of action ahead of your opponent. These feedback loops imply that several desirable characteristics of a cybermodel are:

- Rapid feedback with minimal work to encourage exploration of new configurations, attacks, and defense techniques.
- Rapid prototyping without risk to real systems or data, allowing us to ask "what if" without risk.
- Visualization of the attack as it proceeds to enhance the learning process. When an administrator plays the role of an attacker, it allows the administrator to "think like an attacker" and anticipate how the network is likely to be attacked.

What are the model requirements? The list below is a step toward focusing future research.

- The modeling tools must support rapid learning. Researchers must be able to learn more from the models in a short period of time than we would learn by actually building the network. We must also be able to learn more per dollar using the modeling tool than using a real network.
- The modeling tools must support the prediction of vulnerabilities. For example, the number of bugs in sendmail over the past 10 years could be used to generate a probability that an adversary will find and exploit a new vulnerability during the next 6 months.
- The model must be capable of representing both capabilities and objectives of attackers and defenders. The defense against an attacker launching a denial of service attack is significantly different than defending against an attacker attempting to covertly steal secret data.
- The model must allow security researchers to predict adversary behavior as the adversary learns and as the network evolves.
- Modeling tools must allow evaluation of tactical courses of action with enough fidelity that we are not surprised during tactical operations.
- Modeling tools should allow us to look out far enough into the future that we can predict technology needs early enough to develop solutions. We must be proactive rather than simply reactive.

3. Questions and Issues to be Considered

In addition to examining the questions already raised, we propose exploring the following set of questions to advance the state of security modeling research.

- A critical aspect of modeling is to select the right elements to include in the model and to leave out elements that do not contribute significantly; What are the critical elements to include in a security model? Confidentiality, availability, and integrity have been proposed as fundamental elements. Others have suggested value, loss, and time. Fundamental elements must be measurable. Availability is measured today. However, what are the metrics for confidentiality and integrity? Are there laws that relate the fundamental elements to each other?
- How will the model account for the order and sequence of events?
- Will modeling tools ever provide information we don't already know? Put another way, could a computer driven model defeat a human security expert in competition the way Deep Blue defeated chess champion Garry Kasparov?
- What feedback helps the tactical defenders (administrators) improve their performance? Can these models be turned into course of action tools?
- How can we structure the model to encourage innovation instead of constraining us to think only about today's approaches?
- Does the model require that we make assumptions about reality? Could attacks against the real system be developed by invalidating these assumptions?

4. A Strawman Approach to Modeling COTS

Our DARPA/AFRL sponsored modeling research is mixing and matching the most desirable features from several classes of models. By extracting the operation of real world networks and their vulnerabilities we hope
to create a model that mimics the important elements of the network while minimizing the unnecessary elements. Some differences between our strawman model and earlier security modeling efforts are;

- **Active vs. Static**. The static languages of earlier models limited the types of tools and analysis that could practically be performed. Object oriented models can provide the richness necessary to model and analyze distributed systems. The layered object based Cybermodel will contain code allowing active simulation instead of simply static analysis.
-
- **Custom vs. COTS**. Previous security modeling methods were aimed at custom built systems in which the designer developed the system specifications and had control over the source code. The cybermodel will allow for functional (including vulnerabilities) models of COTS components where the model builder does not control source code or specifications for the COTS product.
-
- **Greater Composability**. Previous efforts were not easily composable. Adding a new element or combining elements often required manual rewriting and analysis. The cybermodel will allow components to be plugged together to create larger, more complex systems. This shadows the way actual distributed systems are constructed.

Figure 4 below provides a *grossly* simplified view of a single node represented by a layered service model object. Each layer receives services from the layer below it and provides services to the layer above it, much like a protocol stack. Multiple nodes can be linked via the network services stack.

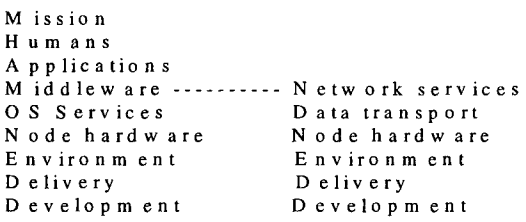

Figure 4 The layered service model for a single node.

The strawman process for using the cybermodel is outlined below.

1. Perform auto-discovery to learn the configuration of any existing components or infrastructure.
2. Manually enter information on new components.
3. Populate the model with component models from the component library. Each node is configured using the information from steps 1 and 2. Nodes are connected using the network configuration information.
4. Overlay vulnerabilities extracted from the vulnerability database taking into account the configuration information for each node.
5. Perform analysis and testing.

5. Acknowledgments

The security modeling work being performed at Secure Computing is being funded under DARPA/AFRL Contract # F30602-97-C-0245, Information Assurance. The author is grateful to all those at the New Security Paradigms Workshop that provided valuable insight and feedback.

6. References

The Collapse of Chaos, Jack Cohen, Ian Stewart, Penguin Publishing, 1995

A Guide to Understanding Security Modeling in Trusted Systems, National Computer Security Center, October 1992, NCSC-TG-010.

Managing Network Security, Computer Games and Network Security Management, Fred Cohen, Journal of Network Security, April 1997

Modeling Information Warfare Effects On Information Operations, A Synopsis Of The Information Operations And Warfare Assessment (Iowa) Simulation, DR. Paul E. Girard, SAIC, 10260 Campus Point Drive, San Diego, CA 92121, August 26, 1998

The Search for Solutions, Horace Freeland Judson,

The Use of Object-Oriented Analysis Methods in Surety Analysis, Gregory D. Wyss, Richard L. Craft, Donald R. Funkhouser, Sandia National Laboratories, SAND99-1242, May 1999

Wargames Revival Breaks New Ground, William B. Scott/FT. Monroe, VA., and Maxwell AFB, VA. Aviation Week & Space Technology, November 2, 1998, The McGraw-Hill Companies Inc.