Proceedings of the 2013 Winter Simulation Conference
R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, eds.

ARCHITECTURE-BASED NETWORK SIMULATION FOR CYBER SECURITY

John A. Hamilton, Jr.

Office of Research
Mississippi State University
P.O. Box 6343
Mississippi State, MS 39762

ABSTRACT

An "executable architecture" is defined as the use of dynamic simulation software to evaluate architecture models (DOD AFWG 2004). By modeling an existing network in the form of an "as-is" architecture, we can create a simulation model, which when stimulated with appropriate traffic, can be an executable architecture.

The DOD Architecture Framework (DODAF) prescribes a modeling framework to capture high-level system design and operational requirements. The system attributes from a DODAF-compliant architecture can directly load a network simulator (Hamilton 2006).

The use of network simulation to study denial of service attacks is well known. However, modeling and simulation techniques can be used to evaluate intrusion detection systems, place and configure security appliances and to design appropriate access control mechanisms.

This paper will discuss the enabling technologies necessary to mainstream architecture-based network simulation including visualization of security requirements, auto generation of network architecture artifacts and application of stochastic elements to the architecture.

1 INTRODUCTION

A soldier standing on a pile of sand in Southwest Asia attempts to send message traffic from his COTS laptop to his higher headquarters using a satellite card. On a good day, he has limited connectivity. When the soldier applies all the regulation-mandated security controls; he transitions from limited connectivity to no connectivity. The simple answer to this dilemma is that the commander makes a decision based on the situation. However, it is clear that there is a need to evaluate the performance costs associated with prescribed security appliances.

The DOD Architecture Framework is the prescribed methodology for documenting system connectivity (CJCSI 6212 2012.01F, CJCSI 3170.01H 2012, DODI 5000.02 2008). The mandatory use of the DOD Architecture Framework is prescribed in DOD Instruction 5000.02, in which responsibility for operational views is assigned to the Joint Staff, while the Under Secretary of Defense (Acquisition, Technology, and Logistics) (USD (AT&L)), leads the development of the system views in collaboration with the Services, Agencies and Combatant Commanders. It is reasonable to use the DODAF artifacts mandated during the acquisition process to evaluate the feasibility of proposed security solutions.

This paper will present a case study modeling a notional airline reservation system using the DODAF methodology. Then we outline how the resulting model can support a network simulation that evaluates security architecture. The case study is based on work performed by Dr. Mark Kuhr and Dr. Derek Sanders while they were students at Auburn University under the supervision of the author. The author has made minor adaptions to illustrate the security architecture requirements.

2 REQUIREMENTS TO CONNECT: THE OPERATIONAL VIEWPOINTS

As defined in the DOD Architecture Framework, an Operational Viewpoint (OV) "is a description of the tasks and activities, operational elements, and information exchange required to accomplish DOD missions." These viewpoints can be used to determine connectivity requirements. Default-deny is a well-known and effective security strategy. Implementing this strategy can be resource intensive and heavy handed. How do you decide what access is appropriate in a given scenario? The DODAF operational viewpoints can be adapted to visualize these security requirements. In figure 1 we see the communication nodes for an airline reservation system are defined. The needlines (labeled NDL) illustrate the requirement for two nodes to exchange information. The needlines are further labeled with the information type and the activity supported. In this example the nodes are further labeled with the Mission Activity Code (MAC) category as defined in DODI 8500.2. There are three defined MAC levels, MAC 1 being the highest priority and MAC 3 being the lowest.

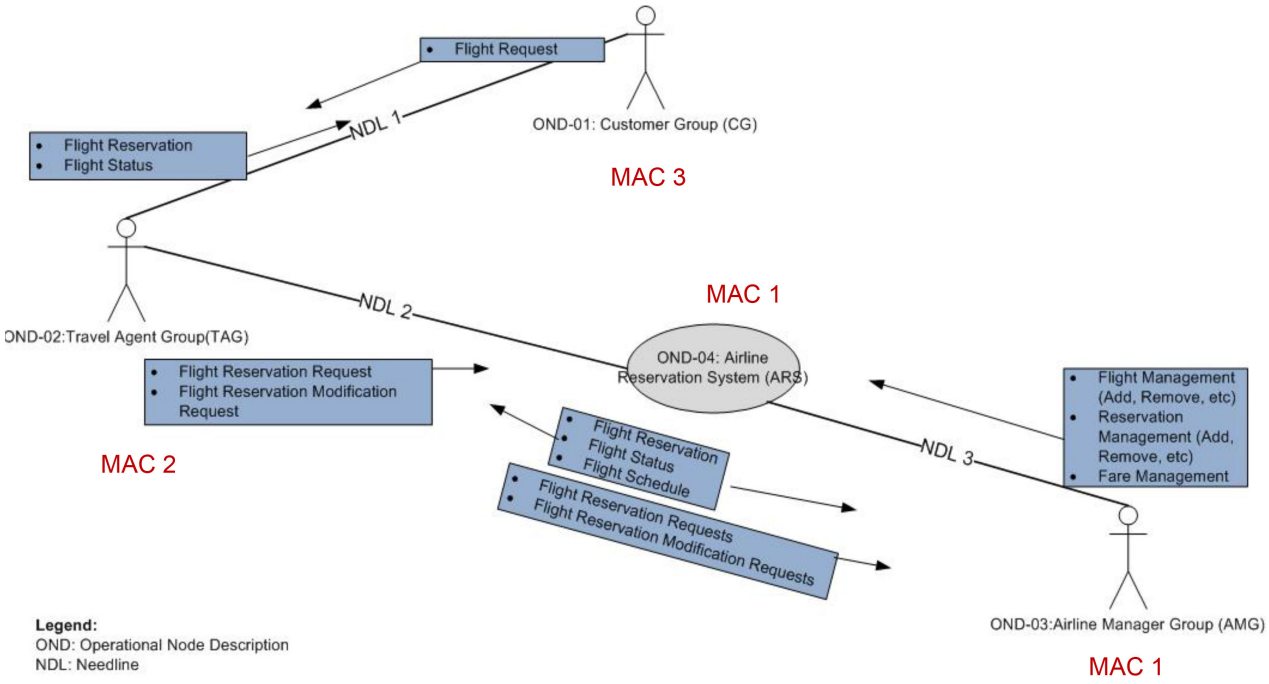

Figure 1: OV-2 Node Connectivity Diagram (Kuhr, Sanders and Hamilton 2008)

In Figure 2 the details of information exchange elements are listed in the OV-3 Operational Information Exchange Matrix. The matrix in Figure 2 is an extract of the complete matrix. The level of detail is sufficient to support high-level design. Labeling the security requirement of each information element provides useful design information at a granularity level that is not often available during system design. In the example matrix in Figure 2 we can see that each information element across each needline is documented.

Information Exchange Identifier	Information Element Description					Producer		Consumer		Nature of Transaction					
	Information Element Name and Identifier	Content	Scope	Accuracy	Language	Sending Op Node Name and Identifier	Sending Op Activity Name and Identifier	Receiving Op Node Name and Identifier	Receiving Op Activity Name and Identifier	Mission/Scenario	Transaction Type	Triggering Event	Interoperability Level Required	Criticality	Confidentiality Level
CG – TAC 1	Flight Request	Final Destination and time	Global	1 hour	English	CG	Provide travel requirements	TAC	Flight Reservation Request	Make reservation	Collaborate	Customer Request	None	High	Public
CG-TAG 2	Flight Reservation	Flight time, date, location; Passenger information	Global	1 hour	English	TAG	Provide travel arrangements	CG	Receive Flight Reservation	Make reservation	Direct	ARS Response	None	High	SBU
CG-TAG 3	Flight Status	Flight time, date, location	Global	15 Minutes	English	TAG	Provide Flight Status	CG	Receive Flight Status	Check Flight Status	Direct	ARS Response	None	High	Public
TAG-ARS 1	Flight Reservation Request	Final Destination and Time; Passenger information	Global	1 hour	English	TAG	Make Reservation	ARS	Flight Reservation Request	Make reservation	Collaborate	Travel Agent Request	None	High	SBU
ARS – TAG 1	Flight Reservation	Flight time, date, location; Passenger information	Global	1 hour	English	ARS	Confirm Flight Reservation	TAG	Issue Flight Reservation	Make a reservation	Direct	AMG Approves Reservation	None	High	SBU
ARS – TAG 2	Flight Status	Flight time, date, location	Global	15 Minutes	English	ARS	Provide Flight Status	TAG	Send Flight Status	Check Flight Status	Direct	Airline Response	None	High	Public
ARS – TAG 3	Flight Schedule	Flight Id, time, date, location	Global	15 Minutes	English	ARS	Provide Flight Schedule	TAG	Find Flights	Make a reservation	Direct	Airline Response	None	Medium	Public
	Flight	Final Destination and Time;													

Figure 2: OV-3 Operational Information Exchange Matrix (Kuhr and Sanders 2008)

We do not typically think of organizational charts as part of system design. However from a cyber security standpoint, an organization chart can help determine who has a need to know the sensitive information identified in the OV-3. It is a simple matter to add a column to each information element detailing the classification or security sensitivity of each element.

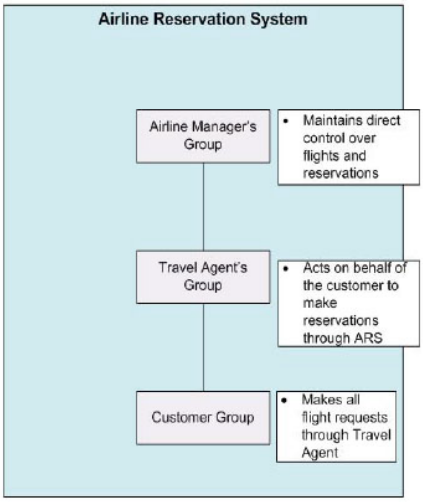

Figure 3: OV-4 Organization Relationship Chart (Kuhr and Sanders 2008)

In Figure 3, the organization chart is based on the nodes identified in the OV-2 and the organization shows what data stores need to be accessed by what organizational nodes. A UML style activity diagram, OV-5 is shown below in Figure 4.

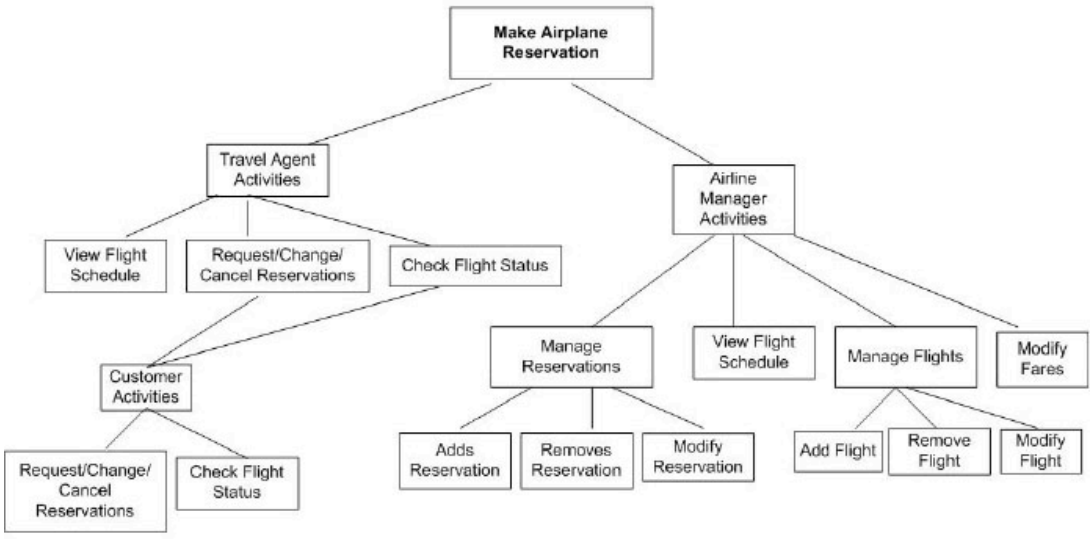

Figure 4: OV-5 Operational Activity Model (Kuhr and Sanders 2008)

The operational viewpoints are a mature methodology for illustrating requirements. By leveraging this methodology, we can make intelligent decisions about what connections to allow and what connections not to allow. By the same token, appropriate access controls can also be visualized. OVs have their limitations. Simply showing requirements does not equal system implementation. However, by evaluating the system against the OVs, we can determine the appropriate types of access and security appliances. Using the DODAF methodology that brings us to the system viewpoints (SVs).

3 HOW THE CONNECTIONS ARE MADE: THE SYSTEM VIEWPOINTS

As described in the DOD Architecture Framework, "A Systems Viewpoint (SV) is a set of graphical and textual products that describes systems and interconnections providing for, or supporting, DOD functions. The SV associates systems resources to the OV." In this regard, several of the key system viewpoints are tied directly to the operational viewpoints. In Figure 5 we see the systems communication description that is built upon the SV-1 (not shown). Each node in the OV-2 must be represented as one or more systems in SV-1/SV-2. The SV-2 builds on the SV-1 and shows how connections are realized i.e. plain old

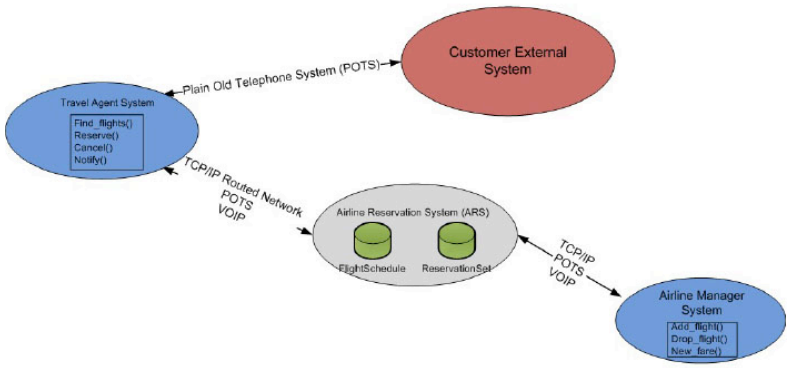

Figure 5: SV-2 System Communications Description (Kuhr, Sanders and Hamilton 2008)

telephone system, for example. Of particular interest in Figure 5 is that the enabling software modules are modeled as systems within the DODAF viewpoint.

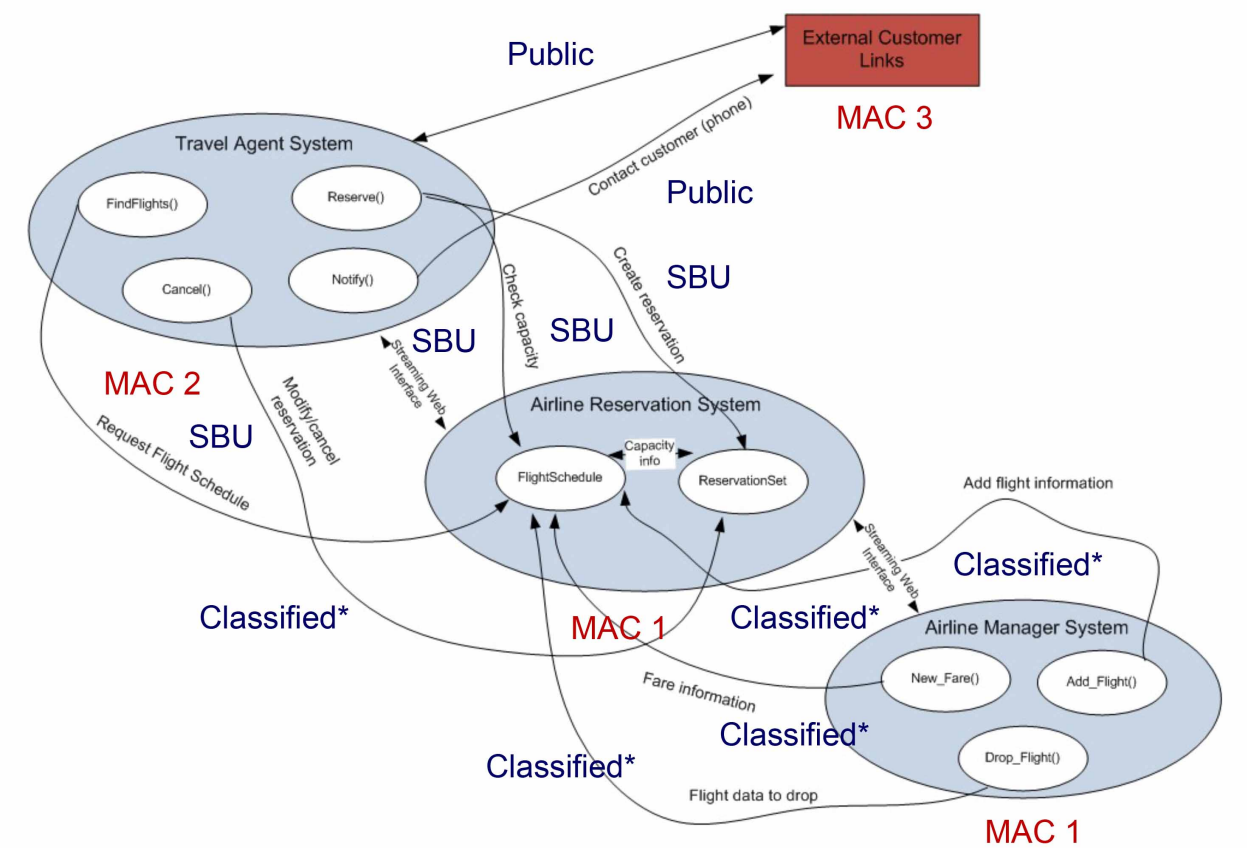

Figure 6: SV-4 System Function Description (Kuhr, Sanders and Hamilton 2008)

Each system is decomposed into functions in the SV-4 System Function Description as shown in Figure 6. Again we go back to the requirements in the operational viewpoints. Just as we mapped nodes from the OV-2 to the SV-2, we now map the system functions described on the SV-4 against the activity diagram in the OV-5 to produce an SV-5, an extract of which is seen in Figure 7. Of interest in the SV-4 is that each function is tied to a software module. We then see in the SV-5 a clear mapping of activities to functions. One obvious use of this documentation is tracing which software modules support which activities. If, for example, the FAA were to mandate new reservation procedures for airlines, then an accurate SV-5 could be useful in identifying which software modules will be affected by the change.

Each system is decomposed into functions in the SV-4 System Function Description as shown in Figure 6. Again we go back to the requirements in the operational viewpoints. Just as we mapped nodes from the OV-2 to the SV-2, we now map the system functions described on the SV-4 against the activity diagram in the OV-5 to produce an SV-5, an extract of which is seen in Figure 7. Of interest in the SV-4 is that each function is tied to a software module. We then see in the SV-5 a clear mapping of activities to functions. One obvious use of this documentation is tracing which software modules support which activities. If, for example, the FAA were to mandate new reservation procedures for airlines, then an accurate SV-5 could be useful in identifying which software modules will be affected by the change.

	View Flight Schedule	Req/Change/Cancel Res.	Check Flight Status	Add Reservations	Remove Reservations	Modify Reservations	Add Flights	Remove Flights	Modify Flights	Modify Fare	Add Fare
Find_Flights()	x										
Cancel()		x			x	x					
Notify()		x	x			x			x	x	
Reserve()		x		x		x					
Flight_Schedule	x		x				x	x	x	x	x
Reservation_Set		x		x	x	x					
New_Fare()										x	x
Add_Flight()							x		x		
Drop_Flight()								x	x		

Figure 7: SV-5 System Function to Operational Activity Mapping

At this point we have cross-walked three ties between the requirements shown in the operational viewpoints and the systems outlined in the systems viewpoints: OV-2 to SV-2, OV-3 to SV-6 and OV-5/SV-4 to SV-5. A major criticism of the DODAF is that is an essentially static series of line and box diagrams. There is some truth to this. Performance insights may be gained from the SV-7 System Performance Measures Matrix as shown in Figure 9. When done well, a list of relevant performance measures can provide a lot of insight into potential performance costs of security appliances.

As stated in the DODAF, a Systems View (SV) is a set of graphical and textual products that describes systems and interconnections providing for, or supporting, DOD functions. The SV associates systems resources to the OV. This association can provide critical insights into the placement of security appliances as well as the performance issues associated with their placement. To fully evaluate these performance issues, simulation is needed.

SV-6 (System View) Airline Reservation System

Interface Indentifier	Data Exchange Identifier	Data Description	Transmission Type	Data Types	Producer	Consumer	Timeliness	Criticality	Security
Travel Agent Interface	new customer information, Flight Schedule Information	Information on which customers are being added to or removed from a flight, Flight information sent back	Modem	String, Boolean, double, TIME, DATE	Travel Agent Interface using all class information for sending and receiving	Flight Booking Interface, Database: using all class information for sending and receiving	Minutes	Medium	SBU
Flight Booking Interface	Connection-Modem	Information about all flights currently scheduled	Modem	String, Boolean, double, TIME, DATE	Flight Booking Interface using all class information for sending and receiving	Database using all class information for sending and receiving	Minutes	Medium	SBU
Airline Manager Interface	Connection-Modem	Information about flight changes and additions	Modem	String, Boolean, double, TIME, DATE	Airline Manager Interface using all class information for sending and receiving	Flight Scheduling, Database: using all class information for sending and receiving	Minutes	Medium	SBU
Flight Scheduling Interface	Connection-Modem	Information about flight changes and additions	Modem	String, Boolean, double, TIME, DATE	Flight Scheduling Interface using all class information for sending and receiving	Database using all class information for sending and receiving	Minutes	Medium	SBU
Data Base	Connection-Modem	Information about flights that are being added or deleted to the schedule	Modem	String, Boolean, double, TIME, DATE	Database using all class information for sending and receiving	Flight Scheduling Interface, Flight Booking Interface: using all class information for sending and receiving	Seconds	High	SBU

Legend

SBU - Sensitive but Unclassified

Figure 8: SV-6 System Information Exchange Matrix (Kuhr and Sanders 2008)

	Performance Range (Threshold and Objective) Measures			
	Time (Baseline Architecture Time Period)	Time	Time (Target Architecture Time Period)	
Airline Reservation System				
Airline Reservation System Hardware				
Maintainability	High	High	High	
Availability	95.00%	97.00%	99.99%	
System Initialization Time	5 Minutes	4 Minutes	2minutes	
Architecture Data Transfer Rate	1GB/sec	1.5 Gb/sec	2 GB/sec	
Program Restart Time	3 Minutes	2 Minutes	1 Minute	
S/W Element 1: Travel Agent Interface				
Architecture Data Capacity (Throughput)	300/sec	500/sec	1000/sec	
Automatic Processing Responses	50.00%	75.00%	95.00%	
Operator Interaction Response Time	30 ms	20ms	5ms	
Availability	99.00%	99.99%	99.99%	
Effectiveness	99.00%			
Mean Time Between S/W Failures	90 days	180 days	360 days	
Organic Training	Yes	Yes	Yes	
S/W Element 2: Airline Manager Interface				
Architecture Data Capacity (Throughput)	300/sec	500/sec	1000/sec	
Automatic Processing Responses	50.00%	75.00%	95.00%	
Operator Interaction Response Time	30 ms	20ms	5ms	
Availability	99.00%	99.99%	99.99%	
Effectiveness	99.00%			
Mean Time Between S/W Failures	90 days	180 days	360 days	
Organic Training	Yes	Yes	Yes	

** **Note: This product will be updated throughout system lifetime**

Figure 9: SV-9 System Performance Measures Matrix

4 EXECUTABLE ARCHITECTURES

As previously noted, an "executable architecture" is defined as the use of dynamic simulation software to evaluate architecture models (DOD AFWG 2004). The system attributes from a DODAF-compliant architecture can be used to directly load a network simulation tool thus producing an executable architecture. Such an executable architecture can be used to validate the operational and system views and check the internal self-consistency of the DODAF compliant architecture.

By modeling an existing network in the form of an "as-is" architecture, we can create a simulation model, which when stimulated with appropriate traffic, can be an executable architecture. One practical example of using executable architectures to support operational planning involves defending against distributed denial of service (DDoS) attacks. A denial of service attack floods a network with so much traffic that legitimate traffic is blocked. This is analogous to jamming a radio network. A distributed DoS attack is one that is launched from many stations instead of a single station. (Mirkovic and Reiher 2004) classify DDoS defense mechanisms as preventive, reactive, cooperation degree and deployment location. An executable architecture can be used to evaluate each type of mechanism. One prevention strategy is to place "forward deployed" firewalls on the outbound ports of the main routers as described in (Chatam 2004). The performance impacts of various firewall configurations and placements are readily displayed though an executable architecture. A typical reactive strategy is to simply reconfigure the network and reroute traffic to a server that is (hopefully) not under a DDoS attack. One autonomous means to mitigate a DDoS attack is to use a dual-queue system, which automatically starts dropping traffic that comes from untrusted hosts at the onset of an attack (Fletcher and Eoff 2004). All of these partial solution strategies to defend against DDoS attacks can be systematically evaluated through an executable architecture.

5 CONCLUSION AND FUTURE WORK

At Mississippi State University, we are working with monitoring and simulation tools developed in the research community to develop automated architecture builders with direct feeds into network simulators. In many cases source code is readily available which provides the capability for extensibility and better understanding.

We need new ways to evaluate assurance of the line-of-attack and point defense components of overall security architectures. The methodologies would consider the entire security architecture (including network and perimeter defense), as part of the consideration of how much assurance is needed at the line-of-attack and point defense level. While a lot is known about development and fielding of secure software-intensive systems built by vetted developers in highly secure environments, it is well recognized that cost and time-to-market are important issues for such developments. Recognizing that line-of-attack and point defense solutions are tightly coupled to the application software in a system, practical strategies are needed regarding when to apply added assurances and when not. In addition, a broader array of assurance solutions and corresponding support tools are needed to provide a larger set of alternatives regarding assurance levels.

It should be noted that DOD programs routinely utilize software as well as hardware subsystems from a variety of unvetted sources. Furthermore, even when development processes and evaluations are guided by recognized methods such as DIACAP and the Common Criteria, some low-level vulnerabilities inevitably escape detection. What is needed is the capability to auto-generate software architectures that can:

1. Verify conformance with proven secure design patterns.
2. Verify conformance to high-level designs specified by the mandated DODAF views.
3. Aid traceability of security requirements and implementation as part of a Security Aware system effort.

An effort to unify DOD information assurance and architecture framework research is the best way to move this research forward. Working across boundaries in this manner is also likely to reduce costs and increase acquisition efficiency.

REFERENCES

Chairman, Joint Chiefs of Staff Instruction (CJCSI) 3170.01H, 2012, *Joint Capabilities Integration and Development System*, Washington, DC.

Chairman, Joint Chiefs of Staff Instruction (CJCSI) 6212.01F, 2012, *Net-Ready Key Performance Parameter*, Washington, DC.

Chatam, J.W., 2004 *Using Strategic Firewall Placement to Mitigate the Effects of Distributed Denial of Service Attacks*, Thesis, Auburn University.

DOD Instruction 5000.02, 2008, *Operation of the Defense Acquisition System*, Washington, DC.

DOD Architecture Framework Working Group 2004, DOD Architecture Framework Volume 1 Definitions and Guidelines, Washington, DC.

Fletcher, H.W. and Eoff, B., "Braving the Storm: Maintaining Connectivity in the Face of a DDoS Attack Using Trusted Hosts," unpublished manuscript.

Hamilton, J. A., Jr. 2006, "A Conceptual Model for Interoperable Command and Control Acquisition." *Journal of Defense Modeling and Simulation,* vol.3, no.2, pp 125-138.

Kuhr, M. and Sanders, D., 2008 Unpublished laboratory work at Auburn University.

Kuhr, M., Sanders, D., and Hamilton, J. A., Jr. 2008 Unpublished laboratory work at Auburn University.

Mirkovic, J. and Reiher, P., "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms," *ACM SIGCOMM Computer Communications Review,* Vol. 34, Number 2, April 2004, pp 39 – 53.

AUTHOR BIOGRAPHY

DREW HAMILTON is Associate Vice President for Research and Professor of Computer Science & Engineering at Mississippi State University. He was previously an Alumni Professor Computer Science & Software Engineering and Director of the Auburn Cyber Research Center at Auburn University. Dr. Hamilton has a B.A. in Journalism/Public Relations from Texas Tech University, an M.S. in Systems Management from the University of Southern California, an M.S. in Computer Science from Vanderbilt University and a Ph.D. in Computer Science at Texas A&M University. Dr. Hamilton is a graduate of the Naval War College with distinction. He is a Past President of the Society for Modeling & Simulation, International (SCS), and Immediate Past Chair of ACM's Special Interest Group on Simulation (SIGSIM) and is on the Board of Directors of the Colloquium for Information System Security Education. He serves as an associate editor for the Journal of Defense Modeling and Simulation as well as the Transactions of the Society for Modeling and Simulation International. Dr. Hamilton's research interests include cyber warfare, digital forensics, cloud computing security, simulation of computer networks, practical applications of the DOD Architecture Framework (DODAF), prevention/protection against distributed denial of service attacks and software vulnerability analysis. His email is hamilton@research.msstate.edu.