



ARL-SR-0349 • DEC 2015



Proceedings of the NATO IST-128 Workshop: Assessing Mission Impact of Cyberattacks

Compiled by Alexander Kott

Approved for public release; distribution is unlimited.

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.



Proceedings of the NATO IST-128 Workshop: Assessing Mission Impact of Cyberattacks

Compiled by Alexander Kott

Preface

These papers were presented during the technical sessions of NATO workshop IST-128 / RWS-019, entitled “Cyber Attack Detection, Forensics and Attribution for Assessment of Mission Impact.” The first technical session focused on the need to gain insights into the intent, motivations, and capabilities of the attackers, in order to understand the intended and actual mission impact. The second explored whether, and to what extent, it is possible to understand the mission impact by analyzing the observable cyber signal and events, through such means as are normally associated with cyber intrusion detection, forensics, and malware analysis. The third discussed the need for models of missions and systems that support missions, and the approaches to constructing such models. The fourth technical session investigated the means by which mission impact could be simulated or modeled. Additional details and a report generated based on the discussions at the workshop are documented in a separate publication titled “Assessing Mission Impact of Cyberattacks: Report of the NATO IST-128 Workshop.”

The Papers of the Workshop IST-128

- 1 Cyber Weapons Development and Testing - Detection, Attribution, Assessment and Deterrence: An Important Challenge to the R&D Community**
by Samuel VISNER, ISF International, USA 1
- 2 Mission Impact and the Role of Behavioral Science**
by Cleotilde GONZALEZ, Carnegie Mellon University, USA 11
- 3 Probabilistic Mission Impact Assessment Based on Widespread Local Events**
by Alexander MOTZEK, Ralf MÖLLER, Mona LANGE, University of Lübeck, DEU, Samuel DUBUS, ALTACEL Lucent, FRA 16
- 4 Software Correlation for Malware Characterization**
by Philippe CHARLAND, Defence R&D Canada, CAN 23
- 5 Estimating Attack Intent and Mission Impact from Detection Signals** by
Patrick McDANIEL, Rober WALLS, Pennsylvania State University, USA 46
- 6 Mission Impact Assessment in Power Grids**
by Mona LANGE, Ralf MÖLLER, University of Lübeck, DEU, Marina KROTOFIL, European Network for Cyber Security 51
- 7 Sensory Channel Threats to Military CPS and IoT Assets**
by Selcuk ULUAGAC, Florida International University, USA 59
- 8 Mission Assurance as a Function of Scale**
by Pierre TREPAGNIER, Alexia SCHULZ, MIT Lincoln Laboratory, USA 62
- 9 Cyber-Attack as a Contest Game**
by Alexander ALEXEEV, Odessa State Ecological University, UKR, Kerry KRUTILLA, Indiana University, USA 66
- 10 Cyber Risk Analysis of CIS-Dependent Missions: A Modeler's Perspective on Preparing for Detecting and Responding to Cyber Attacks for Assessment of Mission Impact**
by Matthew HENRY, David ZARET, Ryan CARR, Daniel GORDON, Johns Hopkins University, USA 75
- 11 Analyzing Mission Impacts of Cyber Actions (AMICA)**
by Steven NOEL, Jackson LUDWIG, Prem JAIN, Dale JOHNSON, Roshan K THOMAS, Jenny McFARLAND, Ben KING, MITRE Corporation, USA, Seth WEBSTER, Brady TELLO, MIT Lincoln Laboratory, USA 80
- 12 Modeling Risk and Agility Interaction on Tactical Edge**
by James R. MORRIS-KING and Hasan CAM, US Army Research Laboratory, USA 87

UNCLASSIFIED

Page 1 NATO/PfP Workshop, June 15-17, 2015

IST-128 Workshop on Cyber Attack Detection, Forensics and Attribution for
Assessment of Mission Impact

*Title: Cyber Weapons Development and Testing – Detection, Attribution,
Assessment, and Deterrence: An Important Challenge to the R&D Community*

Comments to the NATO/PfP Workshop, June 15-17, 2015, Istanbul, Turkey

Samuel. S. Visner,¹ Senior Vice President and General Manager, Cybersecurity, ICF
International and Adjunct Professor, Science and Technology in International
Affairs, Georgetown University

Samuel Sanders Visner
Senior Vice President and
General Manager, Cybersecurity
ICF International
9300 Lee Highway
Fairfax, VA 22031 USA
samuel.visner@icfi.com
Office: 1-703-225-5860 | Mobile: 1-202-255-4308
Fax: 1-888-732-3689 | Skype: 1-202-697-9466
<http://www.icfi.com/services/cybersecurity>

Ladies and Gentlemen, distinguished participants, and fellow presenters:

Thank you for the privilege of allowing me to pose and describe a key cybersecurity challenge, one that will be important to understanding the cybersecurity capabilities and intentions of potential adversaries and of other powers that NATO must consider in its force planning, political, and operational deliberations. This challenge encompasses a broad range of disciplines, including

UNCLASSIFIED

UNCLASSIFIED

Page 2 NATO/PfP Workshop, June 15-17, 2015

potentially physical and behavioral sciences. I describe this challenge as detecting reliably and characterizing accurately foreign cyber weapons development and testing.

This challenge is important indeed to the mission impact focus of this conference and workshop. NATO and its members are faced with rising concerns about the ability of potential adversaries to pose both tactical and strategic challenges – to our ability to conduct effective missions, the industrial capacity that sustains our military effectiveness, and to state sovereignty. The challenges of attack and exploitation detection, attribution, and characterization are common at all levels; these challenges must be met to assess real or intended effects on mission effectiveness, tactically or strategically, as well as to help us understand what mission and strategic responses are appropriate to respond to such attacks, and, if possible, to deter them.

My concern regarding this challenge was catalyzed by a question posed by a related challenge, that of deterring dangerous offensive cyber operations, either on their own or as part of a general attack against a member nation's infrastructure, conducted at a level that could endanger state sovereignty and the lives of citizens. A Defense Science Board effort regarding cyber deterrence underscores the need to understand how cyber deterrence might work, and some of the challenges that lie before us if cyber deterrence is to become an effective reality. At the same time, the DSB's concerns give us the opportunity to study more closely how adversary attacks work, what adversary intentions might be, how serious might be their effects on our own mission capabilities, including counter-force capabilities, and what adjustments in our own doctrine and operational concepts might be appropriate, at all levels. My own thinking about this problem was aided by a comparison of testing, attribution, and assessment of cyber weapons with nuclear weapons, giving us an opportunity to compare an emerging domain with one already well established.

Deterrence theory itself is highly developed, relating as it does to nuclear attack and it might be argued that aspects of nuclear detection, particularly those relating to detection and transparency, might be useful to building a cyber deterrence paradigm, or architecture. In brief,

UNCLASSIFIED

UNCLASSIFIED

Page 3 NATO/PfP Workshop, June 15-17, 2015

nuclear deterrence relies on several key factors, all of which require the ability to gain an accurate of understanding of a potential adversary's capabilities and intentions. These factors include:ⁱⁱ

- A strong understanding of a potential adversary's geopolitical interests, goals, and strategy;
- Accurate characterization of the adversary's operational concepts, as well as its table of operations and equipment;
- Continuous analysis and evaluation of an adversary's capabilities;
- The ability to detect adversary testing activities, and to characterize the results of those test;
- and the ability to detect activities preparatory to an attack, as well as to detect, characterize, and attribute actual attacks.

Alongside these factors is the need for a certain transparency, or the ability to understand the symbolic language used by an adversary, to detect changes in the adversary's intentions, and to signal clearly to an adversary that those changes have been detected. This need for transparency emerged most clearly in 1962 when US President John F. Kennedy and Soviet Premier Nikita Khrushchev, and their countries, confronted each other in a crisis that was only resolved when a communication channel, albeit an informal one, was established that allowed the two leaders to understand each other's intentions and actions, and to signal the details necessary for a resolution of the crisis. Even the movements of American and Soviet warships became part of this dialogue, allowing the two leaders to gauge the other's true intentions and limits.

Our first factor, the need for a strong understanding of a potential adversary's geopolitical interests, goals, and strategy, as well as the need for transparency, represent the need to understand human, political, and organizational behavior. A good deal of work has been done, particularly in the realm of nuclear deterrence, in the political and social science disciplines. This work helps us evaluate continuously the behavioral trends of potential adversaries in the context

UNCLASSIFIED

UNCLASSIFIED

Page 4 NATO/PfP Workshop, June 15-17, 2015

of their broader strategic profile, and it plays an important role in the indications and warning aspect of nuclear deterrence.

The challenge of nuclear deterrence also led to substantial work in the physical sciences, giving us the means to detect and characterize nuclear weapons developmental and testing activities, although it should be noted that there is not a unanimity of opinion regarding the accuracy and timeliness of those means. Nonetheless, there exists reasonable confidence in NATO's ability to detect, attribute, and describe nuclear weapons developmental and testing activities. The Comprehensive Test Ban Treaty, to which many NATO countries subscribe, relies on four streams of data to detect and characterize those tests, including radionuclide, seismic, infrasonic, and hydro acoustic detection and data.

It is true, however, and probably unfortunate that the level of clarity we have been developed to detect nuclear weapons developmental and testing activities, to attribute these activities, and possibly even to detect activities preparatory to an attack, has not been achieved, and it is this challenge I put before our R&D community.

Have developmental activities taken place that we have not detected and characterized? Such a question – the existence of something that we have been able to postulate but not detect – is difficult to answer. However, we might ask: have tests of these capabilities taken place? In this case, we might speculate usefully.

It is my view that the attack on Sony was likely a weapons test. The FBI has indicated with high confidence its assessment that Democratic People's Republic of Korea, or North Korea, was responsible. Why did North Korea conduct an attack on Sony? Was it out of pique, stimulated by what is, by all accounts a fairly wretched movie? Or, did Pyongyang seek to demonstrate to itself and to others that it poses the cyber capability to strike at distances beyond its kinetic reach? In choosing Sony, the US media subsidiary of a Japanese consumer electronics company, it engaged in a scenario unlikely to spark significant retribution, even as it demonstrated its ability to strike at a prominent commercial enterprise equipped with information technology in

UNCLASSIFIED

UNCLASSIFIED

Page 5 NATO/PfP Workshop, June 15-17, 2015

common use globally. It's useful to recall North Korea's sinking in 2010 with a torpedo a South Korean frigate. That incident verified to North Korea to its torpedoes work and demonstrated to potential adversaries another North Korean capability they cannot afford to overlook.

That North Korea tests kinetic weapons on live, adversary targets is something we have seen, in the case of its test of a torpedo, and its live shelling of South Korean villages. However, our understanding of North Korea's cyber operational concepts, and even our ability to distinguish between may be a "merely" malicious action from a test of a capability that might be used in a more disciplined manner, is far from well evolved.

Perhaps a more troubling example is the recent explosion at a German steel plant. This explosion, which caused significant damage, appears to have been caused by the introduction of malware to the industrial control system used in a blast furnace, malware that caused the blast furnace to explode.

Was this a weapons test? If so, it taught its perpetrator quite a lot about the vulnerability of a specific ICS, and without good attribution, the perpetrator was able to conduct this test without fear of significant consequences. Perhaps even more serious, the lack of attribution means that we are not likely to be able to associate this act with the actor's interests, goals, and strategy, nor are we able to engage in the kind of transparent communication necessary to achieve deterrence.

These are troubling developments, made more troubling perhaps by the much lower barrier to entry that exist for cyber warfare than exist for nuclear weapons. The development of nuclear weapons take substantial resources; even their testing and evaluation are complex and resource-intensive activities, allowing in most cases for detection and attribution. Is the same true, however, for cyber weapons? Perhaps not. In fact, probably not.

For the R&D community, much work needs to be done.

UNCLASSIFIED

UNCLASSIFIED

Page 6 NATO/PfP Workshop, June 15-17, 2015

First, what do we know about the real effects of cyber weapons on the systems on which we rely for state sovereignty? What do we know about the effects of these weapons on the infrastructures, damage to which could endanger the lives of our people?

Second, what are the characteristics of a cyber weapons test? What activities take place in preparation for such a test? How are such tests conducted? How are such tests evaluated by those who conduct them? To what extent can we detect such tests and evaluate their results for our own purposes?

Third, are their derivative, or "knock on" effects of such tests that could be detected? In other words, can we detect disturbances in power plants and electrical grids, gas pipelines, steel blast furnaces, and other systems that are indicative of a cyber weapons test or attack, even if the test or attack itself cannot be detected and characterized directly?

Fourth, what are the intended and actual effects of the weapons resulting from such tests on our own mission effectiveness, as well as on our strategic interests?

Fifth, do we understand the behaviors of potential adversaries clearly enough to relate cyber weapons developmental activities and tests to their national interests, goals, and strategies? Do we know what kind of attacks they might be prepared to mount, what are their behavior limits, and what we could hold at risk that they would value enough to deter them from launching an attack?

I would argue that if cyber deterrence is important, then we are as a NATO cyber community in an epoch equivalent to the days in 1962 before the Cuban Missile crisis, an era in which our detection of adversary nuclear activities and our ability to relate those activities to an adversary's behavior, interests, goals, and strategy was sufficiently weak as to place us under the threat of nuclear combat. It is my hope, however, that we won't need an analogous crisis to stir us to action to gain the capabilities necessary to detect, attribute, and characterize cyber weapons tests, to use this understanding to deter their use in a way that threatens our core interests, and

UNCLASSIFIED

to prepare ourselves for a world in which these weapons are increasingly common. In addition, as our own mission effectiveness becomes increasingly a function of our ability to employ complex information systems, our ability to detect, attribute, characterize, and assess adversary cyber weapons test and the employment of these weapons will bear directly on our operational outcomes.

This is the challenge I put before you. Solving it is going to require work in the physical sciences, computer sciences, and behavioral sciences. It will require our best efforts. First steps will include a recognition of the seriousness of this challenge. It will also require closer work with intelligence agencies, helping them define more closely their requirements for collection and analysis. It will also require us to conduct R&D that detects cyber weapons tests directly, as well as the knock-on effects these test might have on critical infrastructures, command and control systems, and other systems. It will also require us to develop the means to characterize human and political behavior of potential adversaries and relate what we observe regarding cyber weapons tests to that behavior.

This won't be easy, but it won't be impossible.

Thank you again, ladies and gentlemen, for your kind attention.

¹ The author is Senior Vice President and General Manager, Cybersecurity, ICF International and adjunct professor of cybersecurity at Georgetown University (Science and Technology and International Affairs Program of the School of Foreign Service). The author served previously as Chief of Signals Intelligence Programs at the National Security Agency.

ⁱⁱ Notes describing differences between cyber and nuclear deterrence, prepared by the author for the Intelligence and National Security Alliance in support of the DSB Cyber Deterrence Study.

Cyber Deterrence (versus nuclear case)

Characteristic	Nuclear	Cyber	Challenges and Comments
Use Attribution	Strong	Generally weak	For cyber, progress is possible; some progress has been made; results remain uncertain

UNCLASSIFIED

Test Attribution	Strong	Generally weak	For nuclear, there remains some debate (e.g., verifiability for purposes of the Comprehensive Test Ban Treaty); testing generally detectable and attributable
			For cyber, lack of consensus on characteristics of what constitutes a test. Possible examples: Sony, Germany steel plant (explosion against industrial control system)
State of the Art	Nuclear weapons and delivery systems well understood	Evolving; difficult to characterize; wide range of attacks and exploits; wide range of delivery vehicles	Range of delivery vehicles (including social engineering) complicates detection and attribution
Developmental Intelligence	We seek to limit intelligence gathered about our own programs. We respect, however, that deterrence relies on some transparency.	Ambiguous US position related to preservation of our capabilities	
Indications and Warning	Well developed; capabilities continue to develop	No consensus	For cyber, wide range of payloads and delivery and low barriers to entry for smaller powers complicates I&W
Standards of conduct (and communication)	Post Cuban Missile Crisis evolution of communication and confidence-building	Lack of standards and communication	Tallinn manual provides first look at cyber warfare codes of conduct; not binding

UNCLASSIFIED

UNCLASSIFIED

Page 9

NATO/PfP Workshop, June 15-17, 2015

Command and control

Well evolved in the US and other
great powers; evolving elsewhere

Poorly evolved globally;
complicates detection

UNCLASSIFIED

NATO STO U.S. National Coordinator
OASD (R&E)/International Technology Programs
4800 Mark Center Drive, Suite 17D08
Alexandria, VA 22350-3600
E-mail: david.r.uribe.ctr@mail.mil
usnatcor@osd.mil

March 30, 2015

Dear Sir,

At the suggestion of Dr. Alexander Kott, I am submitting the attached paper for the Assessment of Impact from Cyber Attack I 2015 : NATO/PfP Workshop on Cyber Attack Detection, Forensics and Attribution for Assessment of Mission Impact IST-128. In accordance with the instructions provided on the Workshop Website, please note the following:

All U.S. Authors must submit one electronic copy to this P.O.C. by 1 APRIL 2015

All US Authors must include the following statement in a covering letter:

- The work described in this paper is cleared for presentation to NATO audiences (i.e., Approved for public release)
- The paper is technically correct
- If work is sponsored by a government agency, identify the organization and attest that the organization is aware of submission. This paper is not sponsored by a government agency.
- The paper is NATO/PfP Unclassified; and
- The paper does not violate any proprietary rights.

Please contact me at Samuel.visner@icfi.com or at 202-255-4308 if you require additional information.

//signed//

Samuel Sanders Visner

Senior Vice President and General Manager, ICF International
and Adjunct Professor, Georgetown University

IST-128 Workshop on Cyber Attack Detection, Forensics and Attribution Assessment of Mission Impact

Mission Impact and the Role of Behavioral Science

Cleotilde Gonzalez

Dynamic Decision Making Laboratory

Department of Social and Decision Sciences

Carnegie Mellon University

5000 Forbes Ave. Porter Hall 208, Pittsburgh, PA, 15213

phone: 412-268-6242

Fax: 412-268-6938

E-mail: coty@cmu.edu

Cyberspace is built by humans to access and share information via information systems, computers, and communications technology. The centrality of human behavior in the cyber world is illustrated in the process assessment of actual and/or intended damage of our missions. Defenders must be aware of possible attackers' behavior and strategies, while they must also understand the technological state of networks and ongoing activities in order to prevent or reduce the negative impact of cyber attacks. In the complexity of cyberspace where the risks of massive damage are high, one cannot forget that the source of all these risks and complexities are humans. At this stage, one should know that arming humans with technology and large amounts of information will not necessarily result in a more secure cyberspace. We must guarantee an integration of our knowledge of human behavior into the development of the wide range of security technology and information filtering efforts. Here I will discuss some challenges (i.e., knowledge gaps) in our understanding of human behavior in the cyber world, which need to be addressed in future research programs. Then, I will present an approach to the integration of computational representations of human behavior and security systems and technology.

Knowledge gaps and research challenges

Cybersecurity issues emerge from a collection of human activities: ill-intentioned and technologically advanced actors in cyberspace, ubiquitous integration and reliance on information and security technology into societal and personal activities, and the need for protection and privacy on property and assets. We need to know more about these human activities and their interaction with technology.

In contrast to the physical world, there are many distinct challenges to human behavior in the cyber world. First, the amount of data available is unusually large and highly diverse. This is due to relatively inexpensive ways of collecting data (e.g., network activity) and to the number and diversity of possible data sources (each network node or piece of equipment can serve as a sensor). Second, cyber attacks can take many forms, and each form might target different parts or services in the network. As such, an attack might be represented in only one data source or in combinations of several data sources, but not in all the data sources at the same time and in the same manner. Thus, the defender needs to expend more effort in searching for and diagnosing information to achieve appropriate defense strategies. Third, the cyber world involves rapid and constant change. In normal day-to-day operation, changes like the maintenance of network equipment, the addition of sub-networks, and changes in services or users may be legitimate operations; however, they may also resemble signs of an attack. Furthermore, changes in

network behaviors can be abrupt, drastic, and caused by both internal and external factors. For example, a sudden spike in network activity on a retailer network can be caused by an approaching holiday (external), the retailer having a sale (internal), or a cyber attack. Fourth, adequate human awareness highly depends on the information coming from sensors (network monitoring equipment, logs, etc.). A defender needs to constantly determine his trust in the sensors and whether or not to rely on the information coming from them; as it is not possible to directly evaluate the sensors' reliability. For example, an attacker may first compromise sensors to deceive a defender about the status of the network before and during the attack. Fifth, cyber attacks are adversarial digital ways of determining who gets power, wealth, and resources. A concept of *Adversarial* awareness needs to be developed to enhance the theory and models of theory of mind in cyber settings. In general, we know little about why humans behave unsafely, how humans might protect their assets from attackers, and how network defenders may learn to predict an attacker's intentions and detect cyber attacks. A program to investigate general basic questions such as how people may learn to protect their own goods while faced with adversaries motivated to steal them, and how the presence of technology and interactions with levels of uncertainty and information may influence attack and defense behaviors may be addressed through the use of security behavioral game theory and the application of economic and psychological models in the investigation of human behavior.

In summary, given the challenges of the cyber world and their implications for human behavior, a parallel research program to investigate the development of computational representations of human behavior can be used to address these gaps.

Computational representations of human behavior and their integration into security technology

In order to create adaptable technology that accounts for the human behavior, cognitive states, limitations, and biases, one needs to ultimately represent these processes in a computational form. Theories of human behavior have been translated at different levels into computational representations. A long tradition of this type of research dates back to the beginnings of Artificial Intelligence and its followers, cognitive architectures. Modeling human behavior in cyber security is challenging, given the gaps identified above, but many efforts are under development. For example, pattern recognition under uncertainty represents a defender's attempt to find patterns in the attacker's action sequence to predict the attacker's next operation and to provide the best response to it. However, if the attacker is aware of these attempts to detect sequential dependencies, one possible path of action is to constantly change the malicious operations and to exploit sequential dependencies. Cognitive models in ACT-R (Anderson and Lebiere, 1998, 2003) and neural networks (West and Lebiere, 2001) are capable of accounting for the human ability to detect sequential dependencies, and they use the perceived sequence to project the next action that an opponent will most likely take in a strategic interaction. Also, cognitive models derived from instance-based learning theory (IBLT) (Gonzalez et al., 2003), a theory of decisions from experience in dynamic tasks, may be used to create cognitive models of the intrusion detection process (Dutt, Ahn, & Gonzalez, 2011).

Relatedly, game theory has been used to model and capture strategies of defenders and attackers in security situations (Pita et al., 2008). Similarly, game theory has been used for decision making in cyber security (Alpcan and Baar, 2011; Grossklags et al., 2008; Lye and Wing, 2005; Manshaei et al., 2013; Roy et al., 2010). However, most game-theoretic approaches to security have some limitations and assume either static game models or games with perfect or complete information (Roy et al., 2010). To some extent, these assumptions misrepresent the

reality of the network security context where situations are highly dynamic and the decision maker must rely on imperfect and incomplete information. To overcome this, recent studies attempt to account for the bounded rationality of human actors, especially human adversaries (Pita et al., 2012). However, this and other game-theoretic approaches still do not fully address cognitive mechanisms like memory and learning that drive the human decision making processes and can provide a first-principled predictive account of human performance, including both capabilities and suboptimal biases. Behavioral Game Theory helps to address some of the limitations imposed by game-theoretic approaches and examine how learning from experience and adaptation to the environment influences decision making and risk taking in cyber security (Gonzalez, 2013). Ongoing efforts aim to scale up cognitive models to study interactions between two or more decision makers in social conflicts like the Prisoner's Dilemma (Gonzalez et al., 2014) and the Chicken Game (Oltamari et al., 2013). However, scaling up models of human cognition to cyber worlds with more than two agents involved is still a challenge (Gonzalez, 2013). A key issue is the need for a better understanding of the role of uncertainty and information availability regarding the attackers. Recent studies examine how the availability of descriptive and experiential information influences interactions in social dilemmas (Martin et al. 2013; Oltamari et al. 2013). The key findings of these studies suggest that information is needed for cooperation to emerge, and that lack of information fostered situations where one decision maker tended to exploit the other.

Summary

Like many other problems in our society (e.g., poverty, crime, drug abuse, etc.), cybersecurity will never be solved once and for all. Instead, we should look for strategies to manage the problem in ways that reduce the costs, losses, and damage to our missions. In this position paper, I propose two ways in which this could be accomplished. The first one is by closing the knowledge gaps in understanding human behavior through long-term, multidisciplinary research programs that address the many facets of the mix between behavior and technology. The second one is through the parallel research on the construction of computational representations of human behavior, which can result in the successful integration of these representations and security technology. Understanding and modeling human behaviors of the attacker is tightly connected with the assessment of actual or intended damage to our missions.

References

- Alpcan, T., & Basar, T. (2011). *Network security: A decision and game-theoretic approach*. New York: Cambridge University Press.
- Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Anderson, J. R., & Lebiere, C. (2003). The Newell test for a theory of mind. *Behavioral and Brain Sciences*, 26(5), 587-639. doi: 10.1017/S0140525X0300013X
- Dutt, V., Ahn, Y.-S., & Gonzalez, C. (2011). Cyber situation awareness: Modeling the security analyst in a cyber-attack scenario through instance-based learning. In Y. Li (Ed.), *Lecture Notes in Computer Science* (Vol. 6818, pp. 281-293). Heidelberg: Springer Berlin.
- Gonzalez, C. (2013). From individual decisions from experience to behavioral game theory: Lessons for cyber security. In S. Jajodia, A. K. Ghosh, V. S. Subrahmanian, V. Swarup,

- C. Wang & X. S. Wang (Eds.), *Moving target defense II: Applications of game theory and adversarial modeling* (pp. 73-86). New York: Springer.
- Gonzalez, C., Ben-Asher, N., Martin, J. M., & Dutt, V. (2014). A cognitive model of dynamic cooperation with varied interdependency information. *Cognitive Science*. doi: 10.1111/cogs.12170
- Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance-based learning in dynamic decision making. *Cognitive Science*, 27(4), 591-635. doi: 10.1016/S0364-0213(03)00031-4
- Grossklags, J., Christin, N., & Chuang, J. (2008). Secure or unsure? A game-theoretic analysis of information security games. In *Proceedings of the 17th International Conference on World Wide Web* (pp. 209-218). New York, NY: ACM.
- Lye, K.-W., & Wing, J. M. (2005). Game strategies in network security. *International Journal of Information Security*, 4(1-2), 71-86. doi: 10.1007/s10207-004-0060-x
- Manshaei, M. H., Zhu, Q., Alpcan, T., Bacsar, T., & Hubaux, J. P. (2013). Game theory meets network security and privacy. *ACM Computing Surveys*, 45(3), 25. doi: 10.1145/2480741.2480742
- Martin, J. M., Juvina, I., Lebiere, C., & Gonzalez, C. (2013). The effects of individual and context on aggression in repeated social interaction. *Applied Ergonomics*, 44(5), 710-718.
- Oltramari, A., Lebiere, C., Ben-Asher, N., Juvina, I. & Gonzalez, C. (2013). Modeling strategic dynamics under alternative information conditions. In R. L. West & T. C. Stewart (Eds.), *Proceedings of the 12th International Conference on Cognitive Modeling (ICCM 2013)*.
- Pita, J., Jain, M., Marecki, J., Ordóñez, F., Portway, C., Tambe, M...et al. (2008). Deployed ARMOR protection: The application of game theoretic model for security at the Los Angeles International Airport. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems: Industrial Track* (pp. 125-132). Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.
- Pita, J., John, R., Maheswaran, R., Tambe, M., Yang, R., & Kraus, S. (2012). A robust approach to addressing human adversaries in security games. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems* (pp. 1297-1298). Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.
- Roy, S., Ellis, C., Shiva, S., Dasgupta, D., Shandilya, V., & Wu, Q. (2010). A survey of game theory as applied to network security. In J. Ralph H. Sprague (Ed.), *Proceedings of the 43rd Hawaii International Conference on System Sciences*. Los Alamitos, CA: IEEE.
- West, R. L., & Lebiere, C. (2001). Simple games as dynamic, coupled systems: Randomness and other emergent properties. *Journal of Cognitive Systems Research*, 1(4), 221-239. doi: 10.1016/S1389-0417(00)00014-0

Carnegie Mellon

**Department of Social and
Decision Sciences**
Porter Hall 208
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213-3890
Fax: (412) 268-6938

March 30, 2015

**NATO STO U.S. National Coordinator
OASD (R&E)/International Technology Programs
4800 Mark Center Drive, Suite 17D08
Alexandria, VA 22350-3600
E-mail: david.r.uribe.ctr@mail.mil or usnatcor@osd.mil
Tel: +1 571 372 6539 / 6538
Fax: +1 571 372 6471**

Dear U.S. National Coordinator:

It is my pleasure to submit the position paper attached to the IST-128 Workshop on Cyber Attack Detection, Forensics and Attribution Assessment of Mission Impact (IST-128-RWS-019) to be held in Istanbul, Turkey, June 15-17 2015.

As per the requirements, and as a US author, I include the following statements:

1. the work described in this paper is cleared for presentation to NATO audiences (i.e., Approved for public release);
2. The paper is technically correct;
3. The work is partly supported by the Army Research Laboratory under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.
4. The paper is NATO/PfP Unclassified;
5. The paper does not violate any proprietary rights

Sincerely,



Professor Cleotilde Gonzalez
Associate Research Professor of Decision Sciences
Director, Dynamic Decision Making Laboratory
Social and Decision Sciences Department
Carnegie Mellon University
412-268-6242
coty@cmu.edu

Probabilistic Mission Impact Assessment based on Widespread Local Events

Alexander Motzek*, Ralf Möller*, Mona Lange* and Samuel Dubus†

* Universität zu Lübeck, Institut für Informationssysteme, Ratzeburger Allee 160, 23562 Lübeck, Germany.

Phone: +49 451 500 5716. Email: {motzek,moeller,lange}@ifis.uni-luebeck.de

† Alcatel Lucent, Bell Labs, Route de Villejust, 91625 Nozay, France.

Phone: +33 160 402 623. Email: samuel.dubus@alcatel-lucent.com

Abstract—Assessing and understanding the impact of scattered and widespread events onto a mission is an ongoing problem. Current approaches employ score-based algorithms leading to spurious results. This paper provides a formal, mathematical model for mission impact assessment. Based on this model we reduce mission impact assessment of widespread local events to a well-understood mathematical problem. Following a probabilistic approach, we present a feasible solution to this problem and evaluate the solution experimentally. We put high care in only using actually available data and kinds of expertise.

I. INTRODUCTION

Modeling dependencies of missions on various involved resources is a novel field of research, which pursues the goal of assessing the influences of local impacts on a higher goal, i.e. a mission. Early approaches use ad-hoc methods for impact assessment involving newly established algorithms

In this work, we take a view from different perspectives towards mission impact assessment. We consider three views from three experts from different expertise and bring them inline towards one well-defined mathematical model. Based on this mathematical model we find a well-understood mathematical problem: In a complex dependency network we find multiple widespread events, whose local effects must be assessed towards a global effect. Using a probabilistic approach, we can benefit from existing, well-defined and well-understood algorithms to solve this problem without returning spurious results.

We focus on actual feasibility of data acquisition and keep manual work to a minimum. We demonstrate and evaluate experimentally that our approach is of linear complexity with the size of application.

The rest of this paper is structured as follows: In Sec. II we develop a mathematical model for mission impact modeling based on views from different experts. Based on this model, we discuss mission impact assessment as a formalized problem, its theoretical complexity and give an experimental evaluation in Sec. III. Being an emerging field of research, we give an overview of related work in Sec. IV. Sec. V gives a conclusion and outlook to future work.

II. DEPENDENCIES AND IMPACTS

In the following, we take a view from different perspectives towards mission impact assessment. We consider three views from three experts from different expertise and bring them inline towards one well-defined mathematical model. Based on

this mathematical model we find a well-understood mathematical problem that assesses a mission impact from widespread local events.

Every expert defines a different dependency model, where every modeled entity represents a random variable and a dependency between two entities is represented by a local conditional probability.

Remark 1. *The here presented approach was developed in a business focused use case. Instead of referring to missions, we refer to business processes in a company and we use both terms interchangeably. Every occurrence of a “business” resource should be adaptable to a “mission” resource.* ▲

A. Mission Dependency Model (Business View)

In the field of business intelligence, a complete company or organization, i.e. a good we aim to protect, is modeled as a conglomeration of *business processes*. Commonly, business processes are modeled using the business process modeling notation (BPMN) and a business process is modeled as a (dependent) collection of tasks. This modeling approach is well accepted and can be found, e.g. in [1], [2], [3]. Fig. 1 shows a sketch of a BPMN model used throughout this paper.

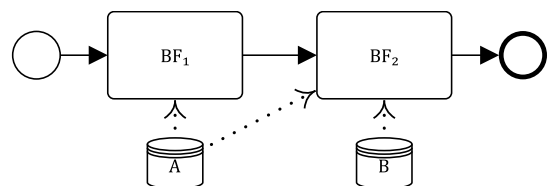


Figure 1. Example BPMN 2.0 model sketch for the BP_1 business process shown in the dependency model of Fig. 2.

Designing BPMN models is handled manually by an expert from a company or by an external business consultant having a precise expertise in the understanding of business analysis. The business analysis is performed on a pure business perspective and stops at a “device” level, e.g. it identifies a web-service, but does not describe the dependencies of the webservice on a database or a data center. This is a reasonable approach, as the latter perspective comes from a very different expertise and would require very broad-range experts. Further, an identification of a “web-service” as a business relevant object is precise in the terms of a business perspective, as, if the web-service is not running, the business process might not be accomplishable. From an “IT” perspective, the web-service

might be irrelevant, as the crucial point of failure lies in the availability of data from a database. The latter dependencies are covered in the upcoming subsection.

We extend [4] and we model mission dependencies as shown in Fig. 2. We model a *company* as being dependent on its *business processes*. A business process is again dependent on one or more *business functions*. *Business devices* provide business function. Business devices are part of the network perspective and—from a network perspective—might be irrelevant, but were identified to be business critical. Fig. 2 shows a dependency graph of business relevant objects, based on the preceding presented BPMN model.

We model every dependency as local conditional probabilities. Every conditional probability describes the probability of failure if a dependency fails. E.g. the probability of the business-function BF_1 (see Fig. 2 and Fig. 1), e.g. “provide access to customer data”, failing, given the required business-device A , e.g. “customer-data database”, is 0.9.

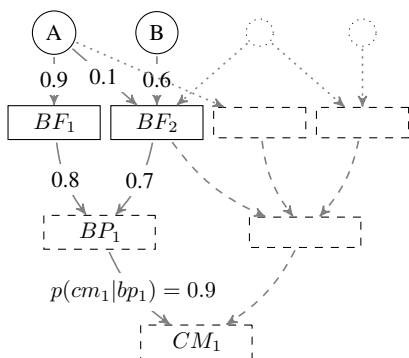


Figure 2. Mission Dependency Model. Values along edges denote individual conditional probability fragments. For our example only the solid entities are used. Consequences and further attributes are omitted in this figure.

Definition 1 (Probabilistic Preliminaries). *We represent every node inside our dependency models by a random variable, denoted as capital X , where every random variable is assignable to one of its possible values $x \in \text{dom}(X)$. Let $P(X = x)$ denote the probability of random variable X having x as a value. For our case we consider $\text{dom}(X) = \{\text{true}, \text{false}\}$ and we write x for the event $X = \text{true}$ and $\neg x$ for $X = \text{false}$.* ▲

The event x represents the case that node X is operationally impacted and $\neg x$ that is is working at its fully operational capacity, i.e. no impact is present.

Definition 2 (From Dependencies to Distributions). *We render every dependency of random variable Y on X as an individual conditional probability $p(x|y)$ and $p(x|\neg y)$. Such individual conditional probability are fragments of a complete conditional probability distribution and are therefore denoted in lower-case. To acquire the local conditional probability distribution $P(X|\vec{Y})$ of node X from all its individual dependencies $p(X|Y)$ of all dependent nodes $Y \in \vec{Y}$, we employ a non-leaky noisy-or combination function [5], [6]. Non-leakiness implies $p(x|\neg y) = 0$ for every dependency and therefore $P(x|\neg \vec{y}) = 0$.* ▲

With Def. 2, we obtain a Bayesian network from our mission dependency model, for which we can specify a joint probability distribution over all entities in the dependency model plainly as the product of all local conditional probability distributions.

The example given below shows how we can use a probabilistic dependency model as a Bayesian network and how an impact can be assessed.

Example 1. *Following the rather simple mission dependency model depicted in Fig. 2 (excluding dashed entities), we obtain the joint probability distribution $P(CM_1, BP_1, BF_1, BF_2, A, B)$ as*

$$= P(CM_1|BP_1) \cdot P(BP_1|BF_1, BF_2) \cdot P(BF_1|A) \cdot P(BF_2|A, B) \cdot P(A) \cdot P(B), \quad (1)$$

where $P(BP_1|BF_1, BF_2)$ and $P(BF_2|A, B)$ are obtained through the noisy-or assumption from $p(bp_1|bf_1)$, $p(bp_1|bf_2)$ and $p(bf_2|a)$, $p(bf_2|b)$ respectively.

We can then marginalize the conditional probability of a mission impact on CM_1 from, say, an observed impact on $A = a$ and none on $B = \neg b$ as

$$P(cm_1|a) = \alpha \cdot \sum_{BP_1} \sum_{BF_1} \sum_{BF_2} P(cm_1, BP_1, BF_1, BF_2, a, \neg b), \quad (2)$$

with a normalizing factor α , s.t. $\sum_{CM_1} P(CM_1|a) = 1$. ♦

To detail an effect of an impact, we define a set of *consequences* for every business process to which a possible failure of the business process might lead. Again, we model a consequence as an individual conditional probability stating the probability that a consequence happens, given an impact on the business process. Likewise, we can then calculate the probability that a BP 's consequence happens (con_{BP}), given all observed local impacts, say a , plainly as $P(con_{BP}|a) = P(con_{BP}|bp) \cdot P(bp|a)$.

Still, only considering a business view does not cover transitively (or passively) involved resources. To cover distant and widespread local events, which are not directly obvious, we introduce a network dependency model in the upcoming subsection.

B. Network Dependency Model (IT View)

As mentioned afore, an identified critical device might be threatened *transitively* by further devices inside the network. In a network modeled by an IT expert we cover dependencies between individual network nodes, which can be, e.g., individual ICT servers, ICS devices, software components or other operationally needed resources. We follow the same “Bayesian” approach as before, i.e. every dependency between two devices represents a local conditional probability of failure, if the dependence fails, as shown in Fig. 3.

However, in contrary to the mission dependency model, assessing network dependencies might not be manageable by hand. Complex network architectures render a manual dependency analysis infeasible and error prone. Further, new dynamically adjusting network architectures make it even

unknown to an expert to identify exact network dependencies. However, it is possible to validate a presented network dependency model for plausibility. We therefore employ heuristics based on exchanged information amounts, e.g. traffic analyses, to identify possible network dependencies. As long as a network device only consumes relevant information for its purpose, every data transfer inside the network must motivate some dependency. Moreover, collecting traffic information about a network is a reasonable and feasible effort. Further, under the assumption of *per node* equally distributed entropy and encoding of consumed information, a dependency, i.e. a conditional probability, must be a function of consumed information bits. We, therefore, reduce an infeasible effort of identifying all dependencies by hand onto finding a heuristic, or rather, checking a generated dependency model.

Example 2. *In our use case, we have information about exchanged information at a logical ICT device level covering virtual machines as individual devices. More granular data, e.g. on software layers, was not acquirable. Fortunately, we can assume in our use case that every device drives one purpose. Multiple software applications running on one device will most likely be dependent on each other, and a failure of one software component will very likely lead to a failure of other software components. We can say that dependencies at device level are coarse enough.*

For example, a workstation X consuming different query results from multiple databases will distribute gained and processed information from such queries to other devices. The percentage of received traffic $T_{Y_i, X}$ from every database Y_i towards the total received traffic can give us a good guideline for the conditional dependency between them as $p(x|y_i) = \frac{T_{Y_i, X}}{\sum_i T_{Y_i, X}}$. However, if the workstation further consumes irrelevant 5TB of cat pictures from a local file server, the heuristic will fail, because the workstation also consumed many irrelevant information. Depending on a network or company characteristics other heuristics might be appropriate, e.g. derivation from a mean received amount of data or a mapping onto a σ distribution. ♦

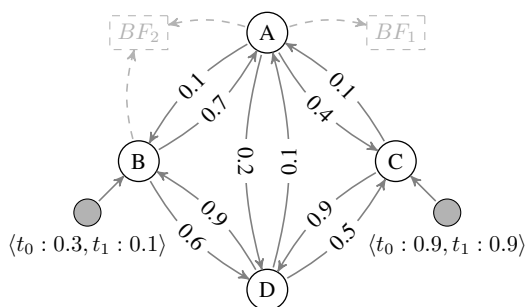


Figure 3. Network Dependency Model. Dependencies between B, C would also be possible. Conditional probability fragments are marked along the edges. Grey nodes represent external shock events leading to local impacts. The time-varying conditional probability of local impact given an instantiated external shock event is given below an event’s node. Connections to the mission dependency model are sketched in dashed grey.

C. Local Impacts (Security View)

A third view involves a security expert able to assess local consequences of events. In the style of reliability analyses using Bayesian approaches we model external shock events inside a network. Every node X might be affected by one or more external shock events \overline{SE} , which are prior random variables. An external shock event $SE \in \overline{SE}$ might be present (se) or not be present ($\neg se$), for which a prior random distribution $P(SE)$ is defined. In the case that an external shock event is present (se), there exists a probability of it affecting a node X , expressed as a local conditional probability fragment $p(x|se)$. If an external shock event exists and it is not inhibited, we speak of a *local impact* on x . In the case that the external shock event is not present, i.e. $\neg se$, it does not affect random variable x and we write $p(x|\neg se) = 0$. Every individual conditional probability fragment from an external shock event is treated in the same noisy-or manner as a dependency towards another node, and thus, multiple shock events can affect one node. We consider fully observable external shock events. Extensions to partially observable shock events are straightforward.

Classically, a local impact can also be seen as an observation of an impacted node, i.e. x . However, in a Bayesian approach this would imply that this impact originates from inside the modeled network and would “blame” other nodes for it. By introducing external shock events we gain the ability to model “soft evidence” of local impacts, i.e. we are not sure whether an external shock event might actually lead to a local-impact and affect a node’s operational capability.

Definition 3 (Temporal Aspects). *We define a temporal aspect of an external shock event. We employ the idea of abstract timeslices in which the effect of an external shock event changes. Every abstract time slice then represents a duplicate of the network- and mission dependencies with a different set of local conditional probabilities of local impacts. We denote time-varying probabilities in a sequence notation as $\langle t_0 : p_0, \dots, t_T : p_T \rangle$, where we have $T + 1$ abstract timeslices. In every abstract timeslice i , varying local impacts take their respective probability p_i defined for its time slice t_i .* ▲

Every local impact represents a potential threat and can be, for example, a consequence of a present vulnerability, a countermeasure, an attack, or originate from hardware failure. It lies in the expertise of a security operator to assess a potential *local* impact of those threats. Note that he does not need to have neither any expertise in network dependencies nor an understanding of missions to do so. The following Ex. 3 shows an example on how external shock events can lead to local impacts in a security context on selecting an adequate response plan to an (ongoing) attack.

Example 3 (Response Plan Side Effects). *We employ mission impact assessment to achieve a qualitative assessment of potential negative side effects of a proposed response plan to an ongoing or potential attack. We see a response plan as a collection of individual actions affecting a network. E.g., a shutdown of a server might easily reduce the surface of a potential attack. Still, if a critical device is highly dependent on that server, it might impact a mission even heavier than a*

potential attack. We consider three mitigation-action types and transform them to external shock events, possibly leading to local impacts.

The first mitigation action, i.e. an external shock event, is a “shutdown”. Obviously, if a node is shut down (se: the external shock event is present) we can easily say that the probability of local impact, given the shutdown of node X , is 1, i.e. $p(x|se) = 1$.

Second, employing a patch on a node X might produce collateral damage as well. During installation of the patch, there might be a (low) probability of immediate conflict. In a mean time, a patch might enforce a reboot of a device. This leads to a temporal shutdown and might lead to hardware failure. Finally, after a successful reboot, a replacement of hardware, and/or a restore of a previous backup, the device will fully resume its operational capability. Using temporal aspects, we can model a patching operation in three abstract time slices and define the local impact probabilities of this external shock event to be $p(x|se) = \langle t_0 : 0.1, t_1 : 1.0, t_2 : 0.0 \rangle$.

Our third considered mitigation action is the restriction of a connection from node X to node Y , i.e. a new firewall rule. From a technical perspective this operation forbids a transfer of data that might have been crucial for the operational capability of a node Y . Therefore, a firewall rule leads to an operational impact on Y . We must assess this impact locally. This is a special case requiring Pearl’s [7] do-calculus. As a connection between two devices resembles a dependency, we must further actually remove this dependency. Otherwise, we would infer further impacts over a dependency that was prohibited and already assessed locally. To do so, we simply “bend” the forbidden dependency to an external shock event se , s.t. the local conditional failure probability $p(y|x)$ becomes a local impact probability $p(y|se)$. Another approach, decidable by a security operator, would be to accumulate dropped connections and add an unified local impact for them. ♦

III. MATHEMATICAL MISSION IMPACT ASSESSMENT

Informally speaking, we have a mission dependency network and a device dependency network. In the device dependency network, some nodes are threatened by external shock events. As nodes are dependent, a threatened node might again threaten another node. We say, a node is threatened by an external shock event *transitively*. This leads to a “spreading” of external shock events. In the end, there exists a probability that even a business process or the complete modeled company (mission) is threatened transitively by various external shock events. To recall, to be threatened by an external shock event (might) lead to an impact; and it is a well-defined problem of calculating this “might”-probability of being impacted due to an external shock event, which is what we call the mission impact assessment.

Definition 4 (Mission Impact Assessment). *The probability of a mission node MN being impacted, is defined as the conditional probability of MN being impacted mn given all observed external shock events $se \in \bar{se}$, i.e. $P(mn|\bar{se})$, where the effects of local impacts due to \bar{se} are mapped globally based on mission-dependency and network-dependency graphs.* ▲

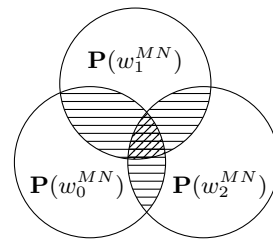


Figure 4. Illustration of $\mathbf{P}(w_i^{MN})$ viewed as sets. Overlapping parts (filled with patterns) are commonly shared probabilities along one path (see Def. 5) are not allowed to be counted twice (or even multiple times) when calculating $\bigcup_i \mathbf{P}(w_i^{MN})$.

Given Def. 4 it is the task of mission impact assessment to calculate the probability $P(mn|\bar{se})$. To calculate this, multiple established approaches are available. From a probabilistic graph view, we need a sound definition of an overall joint probability distribution as demonstrated in Ex. 1. This is well defined for the mission dependency graph, because it is a directed acyclic graph and is a Bayesian network. However, in the network dependency graph we cannot assume an acyclicity constraint and a joint probability distribution is not defined for cyclic graphs. We could therefore, transform the network dependency model into a dynamic Bayesian network and perform a filtering operation on it. However, this introduces a high modeling overhead. Further, we could see the network dependency graph as a Markov random network, which, however, due to a needed global normalization factor, destroys an intended local view on probabilities. Due to the employed noisy-or assumption, we can view the graph and problem as a probabilistic logic program determining the probability of connectivity between a mission node and external shock events. This is a probabilistic path search.

This means to calculate the conditional probability of $P(mn|\bar{se})$, every path w_i^{MN} from an external shock event $se \in \bar{se}$ to the mission node MN is a chain of probabilities and is sufficient to induce $\{MN = true\} = mn$. Every path exists with a probability $P(w_i^{MN})$, where $P(w_i^{MN})$ is the product of all probabilities in this path. Let $\mathbf{P}(w_i^{MN})$ denote the probability viewed as a set. $P(mn|\bar{se})$ is then the probability that at least one path exists. I.e.

$$P(mn|\bar{se}) = P\left(\bigvee_i w_i^{MN}\right) = \bigcup_i \mathbf{P}(w_i^{MN}), \quad (3)$$

where not all $\mathbf{P}(w_i^{MN})$ are disjoint (see Fig. 4) and it is worth noting that all paths might share common “edges”. As every edge represents a probability, plain summation would double count these probabilities and lead to spurious results. This is exactly the issue from which many fudge-factor based “propagation” algorithms in ad-hoc solutions suffer.

An exact calculation of $\bigcup_i \mathbf{P}(w_i^{MN})$ is possible by the inclusion and exclusion principle and the Sylvester-Poincaré equality. Still, calculation is exponentially hard due to the subtraction of all overlapping sets and is therefore not practical. We therefore approximate the result by the use of a Monte-Carlo simulation.

A. Monte-Carlo Approximation

In order to approximate $P(mn|\bar{s}\bar{e})$ we employ a two-step simulation. As discussed in this section we calculate the conditional probability through a probabilistic path search. We first acquire all paths leading to external shock events, for every mission node for which we would like to perform mission impact assessment. Often, we would like to perform this for every node in the mission dependency model. Finding paths for a node in the mission dependency graph is trivial, given found paths from business devices to external shock events. We therefore, as step one, acquire (all) paths leading to evidence for all business devices, which is a classic graph search problem. Under the assumption that the number of business devices and external shock events is comparably small to all nodes in the network graph, a depth-limited search is a reasonable approach for finding paths leading to external shock events.

Definition 5 (Probabilistic Paths). *For every business device $BD_i \in \bar{B}\bar{D}$ let \bar{w}^{BD_i} denote the set of all paths leading to an external shock event and let $w_j^{BD_i}$ denote the j^{th} path. Let \bar{w} denote the super-set of all found paths. Every path $w_j^{BD_i}$ is a set of individual conditional probability fragments $p(x|y)$, representing an edge, i.e. a dependency, from y to x . The product of all probability fragments $p(x|y) \in w_j^{BD_i}$ is the exist-probability of a path $P(w_j^{BD_i})$. Every path $w_k^{BD_i}$ for which holds $\exists j : w_j^{BD_i} \subseteq w_k^{BD_i}$ is irrelevant for calculation and \bar{w} is a finite set. Informally this means, during path search along one path an already visited node must not be visited again and we cannot get stuck in infinite loops. \blacktriangle*

After acquiring all paths \bar{w} leading to all business devices, subsequent paths leading to business functions, processes and the company are trivially acquired by following the paths leading to all children.

Step two is a Monte-Carlo simulation to approximate $P(\bigvee \bar{w}^{BD_i})$ for every business device $BD_i \in \bar{B}\bar{D}$. We draw a sample from \bar{w} and from all dependencies in the mission dependency model. We check for every BD_i the satisfaction of $\bigvee \bar{w}^{BD_i}$ and mark the satisfaction result on BD_i . Subsequently we check for satisfaction of any children, i.e. dependencies, of every node in the mission dependency model. Every satisfaction for a mission node MN found in the mission dependency model is marked as a hit in hit_{MN} . After n_{roll} times, the desired conditional probability of MN being impacted (mn), i.e. the mission impact, given all external shock events $se \in \bar{s}\bar{e}$ is approximated by $P(mn|\bar{s}\bar{e}) = \frac{hit_{MN}}{n_{roll}}$.

Remark 2 (Path Check). *Checking all paths during one Monte-Carlo round is highly optimizable. \bar{w}^{BD_i} can be sorted descending by $P(w_j^{BD_i})$, s.t. most likely existing paths are checked first and subsequent checks can be skipped once a path is found. Further, a path $w_j^{BD_i}$ can be sorted ascending by its individual local conditional probability fragments, s.t. most unlikely random variables are checked first and further checks inside one path can be skipped. Notwithstanding, the complete process is highly parallelizable. \blacktriangle*

Remark 3 (Temporal Aspects Implementation). *We introduced that evidence, i.e. an external shock event, can have different*

conditional local probabilities depending on an abstract time slice. This means we have a varying probability at the end of one path $w_j^{BD_i}$. Naively, we could perform a Monte-Carlo simulation for every abstract time slice. However, this would redundantly simulate all non-varying probabilities. We therefore partition $w_j^{BD_i}$ in a non-varying set of conditional probabilities, i.e. a network path leading to an impacted node, and a set of varying conditional probabilities, i.e. a set of local impacts. \blacktriangle

The following example gives a short demonstration of mission impact assessment using the defined mathematical model using an approximate Monte-Carlo method.

Example 4. *Let us consider Fig. 3, where an identified mission critical device A (compare Fig. 2) is threatened (transitively) by local impacts on nodes B and C . Let us call the local impacts SE_B and SE_C . Let us exclude the dependency of BF_2 on B and temporal aspects for brevity. Through depth-first search we find the paths \bar{w}^A as*

$$\begin{aligned} w_0^A &= \{p(a|b), p(b|se_B)\} \\ w_1^A &= \{p(a|b), p(b|d), p(d|c)p(c|se_C)\} \\ w_2^A &= \{p(a|c), p(c|se_C)\} \\ w_3^A &= \{p(a|c), p(c|d), p(d|b), p(b|se_B)\} \end{aligned} \quad (4)$$

Additional paths, e.g. $w_0^A = \{p(a|b), p(b|c), p(c|b), p(b|se_B)\}$, are redundant, as, here, w_0^A is always (already) satisfied, if w_0^A is satisfied. After finding these paths, finding paths to higher nodes in a mission dependency model, say, to BF_1 , is trivial, by simply appending $p(bf_1|a)$ to every path of A . Subsequently, the same holds for BP_1 and CM_1 .

For the simulation, at first every used random variable is sampled. Let $\vec{R}\vec{V}$ be the vector of all random variables included in all paths. I.e. $\vec{R}\vec{V} = \langle p(a|b), p(b|se_B), p(b|d), p(d|c), p(c|se_C), p(a|c), p(c|d), p(d|b), p(bf_1|a), p(bf_2|a), p(bp_1|bf_1), p(bp_2|bf_2), p(cm_1|bp_1) \rangle$. Let $\vec{r}\vec{v}$ denote a sample of $\vec{R}\vec{V}$, say, $\vec{r}\vec{v} = \langle +, +, +, +, +, -, -, -, +, +, +, +, + \rangle$, where $+$ represents a true sample, and $-$ a false sample.

Subsequently, for every identified critical device, i.e. A , we check if at least one of its path is satisfied, i.e. if $\bigvee \bar{w}^A$ is satisfied. We obtain that w_0^A is satisfied and this satisfies A . The circumstance that w_1^A is also satisfied, but w_2^A and w_3^A are not satisfied is irrelevant and further checks can be skipped. Subsequently, we can check the remaining mission dependency graph for further satisfactions in this sampling round. As A and $p(bf_1|a)$ are satisfied, BF_1 is satisfied as well. The same holds for BF_2 . Likewise, BP_1 is satisfied as well as CM_1 . Every satisfaction is marked as a successful Monte-Carlo round and increments a mission node's MN hit counter hit_{MN} .

This procedure is repeated n_{roll} times, i.e. $\vec{r}\vec{v}$ is sampled and \bar{w} is checked. Finally, every operational impact assessment of a mission node MN , represented by the conditional probability $P(mn|se_C, se_B)$, is approximated by $P(mn|se_C, se_B) = \frac{hit_{MN}}{n_{roll}}$. \blacklozenge

B. Complexity and Experimental Evaluation

We implemented the proposed approach as a flexible framework allowing user defined definitions of local impacts,

user defined heuristics for dependency approaches and user defined performance characteristics as defined below. As central theme, we focused on actual feasibility of our proposal and we demonstrate that our approach scales well, i.e. linear, with a graph's complexity. In the following, we give a short expected summary of the complexity of our approach and evaluate it experimentally.

Evaluation and demonstration of the computational complexity of our presented approach is difficult, as it depends on the graph structure of the network and the processed response plan. We therefore use random graphs containing n_N nodes and $n_E = n_N^2 \cdot 0.1$ edges while assuring that every node is at least bidirected. By doing so we obtain a fully connected graph with, approximately, a 10% chance of two nodes being directly connected. The processed response plan consists of n_{MA} randomly placed mitigation actions (external shock events). For evaluation we place rather many $n_{MA} = n_N \cdot 0.1$ mitigation actions, i.e., 10% of all nodes are possibly impacted. We measure the time t_{ps} required for finding all n_P paths up to depth d_{max} , and t_{sim} required for simulating all found paths n_{roll} times. Every measurement is repeated in 50 different random graphs.

Complexity is differentiated between both steps of the impact assessment. Given a constant maximum search depth d_{max} , depth-limited search (DLS) scales linearly with the number of edges n_E , as also experimentally evaluated in Fig. 5. Further, DLS scales slightly with the number of placed local impacts n_{MA} (compare Fig. 10), as a pre-computation of shortest distances to local impacts per node can eliminate dead ends early. We write, path search is a function as $t_{ps} = f(n_E, d_{max}, n_{MA})$.

Remark 4 (DLS). *Obviously, DLS scales exponentially with a specified maximum depth d_{max} . In general and for our example, the maximum depth should be chosen in the range of the average path length inside a given graph, s.t. almost every node is considered at least once. In order to better scale a maximum depth it is reasonable to allow a rational d_{max} , where a depth $d_{dec} < 1$ resorts to the best $d_{dec}\%$, i.e. most dependent, children.* ▲

Monte-Carlo simulation, i.e. checking of paths, scales linearly with the number of found paths n_P (compare Fig. 6 and 7) and the number of Monte-Carlo rounds n_{rolls} (compare Fig. 8), i.e. $t_{sim} = f(n_P, n_{roll})$. Naturally, the number of proofs n_P scales with the number of local impacts n_{MA} , of edges n_E and the maximum path length d_{max} (compare Fig. 9), i.e. $n_P = f(n_{MA}, n_E, d_{max})$.

In summary, we conclude that experimental results match theoretically expected complexities.

IV. RELATED WORK

Mission modeling and mission impact assessment is an emerging field of research; and, naturally in new, viral research areas, employ ad-hoc solutions using algorithms involving fudge factors. While delivering early results and acclaimed solutions for mission impact assessment, a formal definition of an underlying problem is yet missing. Employed fudge factors

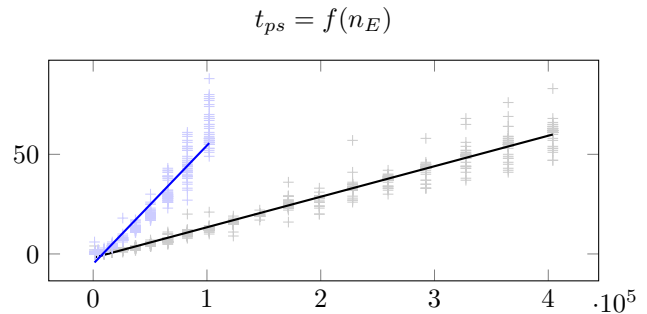


Figure 5. Pathsearch is linear with the number of edges in the graph. Black: $d_{max} = 3$, Blue: $d_{max} = 4$. n_N is linearly increased, meaning a quadratic increase of edges. $t_{ps} = f(n_E, d_{max}, n_{MA})$, n_{MA} only very slightly. t_{ps} in ms.

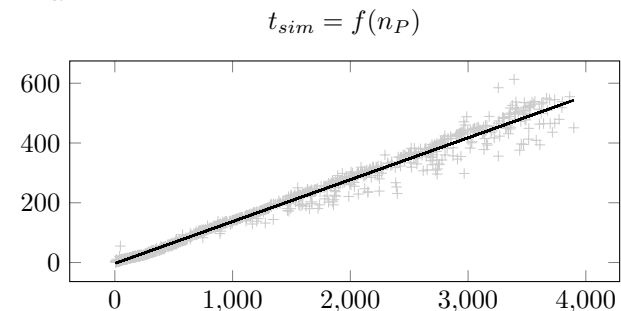


Figure 6. Simulation time is linear with the found paths. $d_{max} = 3$. n_N is linearly increased, meaning a linear increase of mitigation actions and a quadratic increase of edges, both increasing the amount of findable paths. t_{sim} in ms.

in newly established algorithms lead to untraceable and spurious results demanding data driven validations. Unfortunately, large, standardized datasets for validation are yet missing for mission impact assessment and in the following presented work. In the following, we point out valuable approaches and ideas.

Barreto et al. [1] introduce a well-understood modeling technique and use BPMN models to acquire knowledge. An impact assessment is based on various indexes and numerical scores, such as exploit index, impact factor, infrastructure capacity index, and graph distances. Albanese et al. present in [2] a well-modeled formalism for complex inter-dependencies of missions as a set of tasks. Using numerical scores and tolerances in a holistic approach Albanese et al. focus on cost minimization. Buckshaw et al. [8] propose a quantitative risk management by involving various experts and present a score-based assessment based on individual values and a standardization using a weighted sum.

Jacobson [4] presents a well understood conceptual framework using interdependencies based on operational capacity. In this dependency model, impacts are propagated and reduce the operational capacity. [4] uses self-defined metrics for propagating impacts through Boolean gates.

Further works focused solely on modeling. E.g., Goodall et al. [9] focus on modeling and available data integration using ontologies but do not address an impact assessment. Another ontology-based approach is presented by D'Amico et al. in [10] and identifies multiple experts while noting that, e.g., system administrators are not capable of understanding an organization's missions.

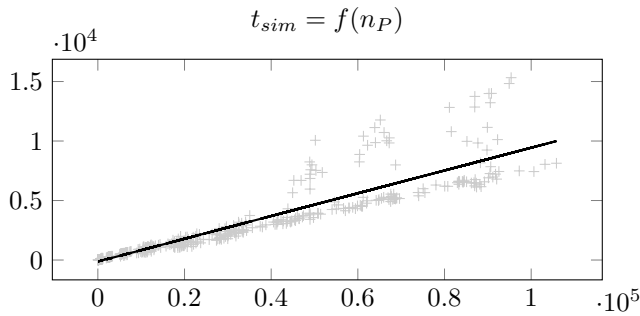


Figure 7. Same simulation as Fig. 6, but for $d_{max} = 4$. Linear time complexity is also achieved for very large proof sets.

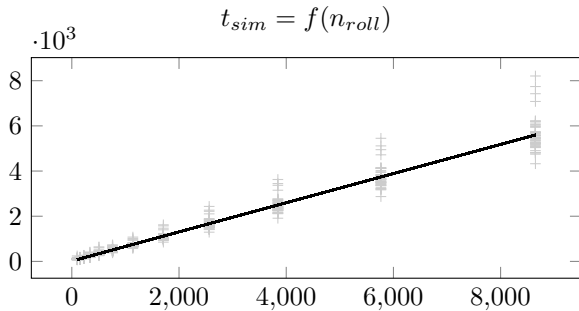


Figure 8. Simulation time directly correlates with the number of rolls. $n_N = 500$, $n_{MA} = 50$, $d_{max} = 4$. Around 12000 proofs are found each time. n_{roll} is increased exponentially and every measurement is repeated for 50 different random graphs. t_{sim} in ms.

Notwithstanding, we were inspired by several aforementioned modeling ideas, such as using the BPMN standard and we considered different views from various experts. To the best of our knowledge, we contribute a novel, formalized, mathematical mission impact assessment to this emerging research area.

V. CONCLUSION

We presented a well-defined mathematical mission impact assessment, based on a probabilistic approach, without introducing score-based propagation algorithms returning spurious results.

We relied on the expertise of different experts and merged all views without losing information or forcing an expert into a knowledge field he cannot understand. Based on an established mathematical model, we reduced mission impact assessment onto a well-understood problem in computer science. Experimental results demonstrate scalability of the approach such that large-scale network scenarios can be handled.

Future work is dedicated to integrating the presented mission impact assessment into a fully automated cyber-defense system.

ACKNOWLEDGMENTS

This work was partly supported by the Seventh Framework Programme (FP7) of the European Commission as part of the PANOPTESec integrated research project (GA 610416).

Approved for public release; distribution unlimited.

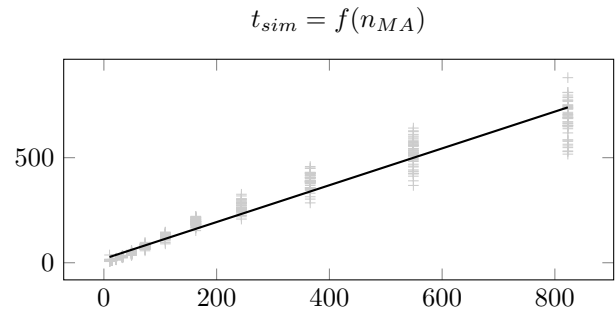


Figure 9. Simulation time directly correlates with the number of mitigation actions. Constant $n_N = 1000$, $d_{max} = 3$, $n_{roll} = 1000$. n_{MA} is increased exponentially and repeated in 50 different random graphs. The more MAs, the more paths, the longer it takes. t_{sim} in ms.

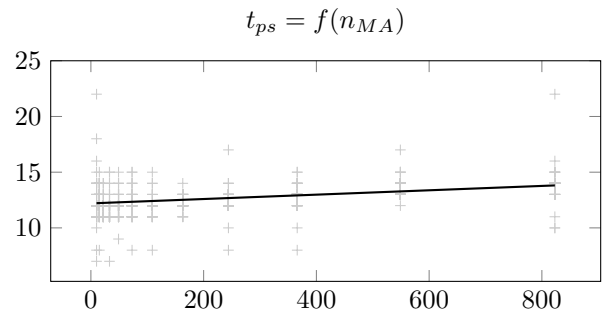


Figure 10. Proofsearch time is negligible dependent of the number of mitigation actions. t_{ps} correlates with the number of edges in the graph. Measurements collected during Fig. 9. t_{ps} in ms.

REFERENCES

- [1] A. de Barros Barreto, P. C. G. da Costa, and E. T. Yano, "Using a Semantic Approach to Cyber Impact Assessment," in *STIDS*, 2013, pp. 101–108.
- [2] M. Albanese, S. Jajodia, R. Jhawar, and V. Piuri, "Reliable mission deployment in vulnerable distributed systems," in *43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop*. IEEE, 2013, pp. 1–8.
- [3] S. Musman, A. Temin, M. Tanner, D. Fox, and B. Pridemore, "Evaluating the Impact of Cyber Attacks on Missions," in *Fifth International Conference on Information Warfare and Security*, 2010, pp. 446–456.
- [4] G. Jakobson, "Mission Cyber Security Situation Assessment using Impact Dependency Graphs," in *Fourteenth International Conference on Information Fusion*. IEEE, 2011, pp. 1–8.
- [5] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 2014.
- [6] M. Henrion, "Practical Issues in Constructing a Bayes' Belief Network," in *Third Conference on Uncertainty in Artificial Intelligence*, 1987.
- [7] J. Pearl, *Causality: Models, Reasoning and Inference*, 2nd ed. New York, NY, USA: Cambridge University Press, 2009.
- [8] D. L. Buckshaw, G. S. Parnell, W. L. Unkenholz, D. L. Parks, J. M. Wallner, and O. S. Saydjari, "Mission Oriented Risk and Design Analysis of Critical Information Systems," *Military Operations Research*, vol. 10, no. 2, pp. 19–38, 2005.
- [9] J. R. Goodall, A. D'Amico, and J. K. Kopylec, "Camus: Automatically Mapping Cyber Assets to Missions and Users," in *Military Communications Conference*. IEEE, 2009, pp. 1–7.
- [10] A. D'Amico, L. Buchanan, J. Goodall, and P. Walczak, "Mission Impact of Cyber Events: Scenarios and Ontology to Express the Relationships between Cyber Assets, Missions, and Users," in *Fifth International Conference on Information Warfare and Security*, 2010, pp. 8–9.

IST-128 Workshop on Cyber Attack Detection, Forensics and Attribution for Assessment of Mission Impact

Software Correlation for Malware Characterization

Philippe Charland

Mission Critical Cyber Security Section
Defence Research and Development Canada – Valcartier Research Centre
2459 de la Bravoure Road, Quebec, QC, G3J 1X5
CANADA
Tel.: 418-844-4000 ext. 4491 Fax: 418-844-4538

philippe.charland@drdc-rddc.gc.ca

ABSTRACT

It is a common scenario that the primary piece of evidence of an attack is the malicious software (malware) used to perpetrate it. Analyzing and characterizing malware requires reverse engineering, a manually intensive and time-consuming process, whose learning curve is quite steep and is hindered when anti-reverse engineering techniques are used. Due to the high technical sophistication required for building advanced and stealthy persistent malware, it is quite common that code fragments are reused, either from other malware, or even from legitimate sources, such as open source code repositories. The analysts should thus take advantage of the code reuse in the production of malware to accelerate the reverse engineering process. This paper presents two assembly code analysis techniques, supported by prototypes, towards this goal, namely assembly to source code matching and assembly code clone search. Using the Citadel and Zeus malware as a case study, these two techniques help to reduce the number of functions which should be manually analyzed by a reverse engineer. The results prove that the approach is promising and is applicable to other malware analysis scenarios.

1.0 INTRODUCTION

Malicious software (malware) presents a direct threat to military operations. With the growing dependence on communications and information systems to support military missions, malware has the capacity to impact the availability of critical system's assets and data, making missions vulnerable to cyber threats. One example is the malware which infected in 2011 the cockpits of America's Predator and Reaper drones at the Creech Air Force Base in Nevada. This malware logged the pilots' every keystroke as they remotely fly missions over Afghanistan and other warzones [1]. It is only by dissecting malware to understand how it works through advanced analysis techniques that it can be defeated or eliminated. If not, then the malware can resist multiple removal efforts and re-infect the systems, such as in the case of Creech's computers [1].

Performing in-depth analysis of malware necessitates reverse engineering, which is not a simple learning endeavour [2]. The learning process is quite involved, as it requires knowledge from several disparate domains, such as computer architecture, systems programming, operating systems, and compilers [2]. Although software reverse engineering came to an age in 1990 [3], it is only in the last decade that its importance and visibility have arisen. However, despite a growing community, it is still perceived by many as a dark art. This paper presents two techniques, together with the prototypes implementing them which, by leveraging open source code repositories publicly available and assembly code fragments previously analyzed, reduce the entry barrier new

Software Correlation for Malware Characterization

analysts face, as well as enhance and accelerate the reverse engineering process. The remainder of this paper is organized as follows: Section 2 presents the assembly to source code matching technique and its associated prototype. Section 3 provides background information on code clone detection, the research area on which the identification of reused code fragments is built, the different types of assembly code clones which should be detected by the proposed technique, as well as the prototype supporting it. Section 4 describes the results of a case study using the two techniques to analyze the Citadel and Zeus malware. Finally, Section 5 concludes the paper.

2.0 ASSEMBLY TO SOURCE CODE MATCHING

Malware authors reuse code from all sources including legitimate ones, such as open source code. For example, both the Conficker worm and the Waledac bot used an open source implementation for their cryptographic functions [4, 5]. In the case of the Waledac bot, this represented 25% of its code [5]. Also, the much-discussed FLAME malware contained publicly available open source code packages such as SQLite and Lua [6].

Another example of source code reuse for malware creation is the Citadel Trojan. Citadel is based on the leaked source code of the Zeus crimeware kit [7]. Other malware (e.g., Gameover Zeus, Ice IX, LICAT, and Murofet) were also created after the source code of Zeus was made public [7].

2.1 Objective

Some people have compared reverse engineering to solving a jigsaw puzzle [8]. You first start by finding the corner pieces, then the frame, and after that, you work your way forward from there. Using this analogy, the corner pieces for reverse engineering are strings, constants, and function names. Strings contain human readable hints about a given functionality. Specific constants can give additional clues and can sometimes be used to even identify certain types of algorithms. Function names of imported functions from shared libraries (e.g., DLL) can reveal information about the performed actions. However, a lot of experience is needed to know, for example, the significance of a constant in a given context or what the combination of imported functions might results in.

The meaning of strings, constants, and function names could be obtained by searching them on publicly available open source code repositories. This was the idea behind the RE-Google IDA Pro plug-in [8], which has proved to be very valuable to find algorithms and code excerpts containing such information on Google Code. The ability to efficiently recognize the open source origin for a given assembly code fragment is desirable, both in order to enhance the productivity of a reverse engineer, as well as to reduce the odds of common libraries leading to false correlation between otherwise unrelated code bases.

RE-Google [8], a proof of concept plug-in for the IDA Pro disassembler [9], extracts constants, library names, and strings contained in a disassembled binary, and uses them to search for code on Google Code [10]. The links to the top ten source code files found are inserted as comments into the assembly code listing. Reviewing these files frequently provides enlightening insights into the functionality of the code fragment in question and saves considerable time. However, RE-Google uses the Google Code Search Data Application Programming Interface (API), which is no longer available, making this plug-in non-functional. Furthermore, Google Code will be shut down in January 2016.

2.2 BinSourcerer Prototype

In order to be able to automatically match assembly with source code, the following alternatives to the Google

Search service were evaluated: Antelink [11], CodePlex [12], GrepCode [13], GitHub [14], Krugle [15], the Open Hub Code Search [16], and searchcode [17]. The two selected options were the Open Hub Code Search and GitHub.

The Open Hub Code Search, by Black Duck Software [18], with its 21,372,664,482 lines of open source code, claims to be the world's largest and most comprehensive code search engine [16]. It is the result of the merge in 2012 with Koders, another open source code search engine acquired by Black Duck Software in 2008. Through the Open Hub Code Search, Black Duck Software wants to fill the gap left by the shutdown of Google Code Search in 2012.

GitHub is a web-based hosting service for software development projects using Git [19] at its heart. Git is a free and open source distributed version control system. GitHub claims to be the largest code host on the planet with over 21.2 million repositories [20]. One advantage that GitHub has over the Open Hub Code Search is its robust API, which allows the integration of third-party tools or applications [20].

BinSourcerer is the prototype implementing the assembly to source code matching technique. It draws its inspiration from RE-Google, but instead of submitting queries to Google Code, it relies on the Open Hub Code Search and GitHub to correlate assembly with source code. BinSourcerer takes as input a target binary file disassembled with IDA Pro and performs the following steps for each function: (i) extraction of interesting features (i.e., strings, constants, and imported function names), (ii) feature-based query encoding, (iii) query refinement for on-line code repository search (i.e., the Open Hub Code Search and GitHub), (iv) request/response processing, (v) data extraction and parsing, and (vi) results reporting.

BinSourcerer has been released as open source code on GitHub¹ under the Apache License, Version 2.0. The following are the different usage scenarios it supports, illustrated with examples.

2.3 Exact Matching

The perfect scenario in assembly to source code matching is when the source code of an assembly code function is found on a public repository. This is illustrated in the following example, where the corresponding source code of the Citadel Trojan function `sub_42514F` (Figure 1) has been found on GitHub (Figure 2).

¹ <https://github.com/BinSigma/BinSourcerer>

Software Correlation for Malware Characterization

```

.text:0042514F ; ----- S U B R O U T I N E -----
.text:0042514F
.text:0042514F sub_42514F      proc near          ; CODE XREF: j_CER_sub_42514F1j
.text:0042514F                                     ; injected_thread_start+CA1p
.text:0042514F      push     ebx
.text:00425150      push     edi
.text:00425151      push     offset szSubsystemProtocol ; "MV"
.text:00425156      push     0             ; hProv
.text:00425158      xor     bl, bl
.text:0042515A      call    ds:CertOpenSystemStoreW
.text:00425160      mov     edi, eax
.text:00425162      test    edi, edi
.text:00425164      jz     short loc_42519A
.text:00425166      push    ebp
.text:00425167      mov     ebp, ds:CertEnumCertificatesInStore
.text:0042516D      push    esi
.text:0042516E      push    0
.text:00425170      jmp     short loc_425185
.text:00425172 ; -----
.text:00425172 loc_425172:      ; CODE XREF: sub_42514F+3D1j
.text:00425172      push    esi             ; pCertContext
.text:00425173      call    ds:CertDuplicateCertificateContext
.text:00425179      test    eax, eax
.text:0042517B      jz     short loc_425184
.text:0042517D      push    eax             ; pCertContext
.text:0042517E      call    ds:CertDeleteCertificateFromStore
.text:00425184 loc_425184:      ; CODE XREF: sub_42514F+2C1j
.text:00425184      push    esi             ; pPrevCertContext
.text:00425185 loc_425185:      ; CODE XREF: sub_42514F+211j
.text:00425185      push    edi             ; hCertStore
.text:00425186      call    ebp ; CertEnumCertificatesInStore
.text:00425188      mov     esi, eax
.text:0042518A      test    esi, esi
.text:0042518C      jnz    short loc_425172
.text:0042518E      push    eax             ; dwFlags
.text:0042518F      push    edi             ; hCertStore
.text:00425190      mov     bl, 1
.text:00425192      call    ds:CertCloseStore
.text:00425198      pop     esi
.text:00425199      pop     ebp
.text:0042519A loc_42519A:      ; CODE XREF: sub_42514F+151j
.text:0042519A      pop     edi
.text:0042519B      mov     al, bl
.text:0042519D      pop     ebx
.text:0042519E      retn
.text:0042519E sub_42514F      endp

```

Figure 1: Citadel sub_42514F function in IDA Pro.

```

168 bool ClearSerts( const char* nameStore )
169 {
170     bool ret = false;
171
172     HANDLE hstore = CertOpenSystemStore( NULL, nameStore );
173     if( hstore != NULL)
174     {
175         PCCERT_CONTEXT certContext = 0;
176         while( (certContext = CertEnumCertificatesInStore( hstore, certContext )) != NULL )
177         {
178             PCCERT_CONTEXT dupCertContext = CertDuplicateCertificateContext(certContext);
179             if( dupCertContext != NULL )
180                 CertDeleteCertificateFromStore(dupCertContext);
181         }
182         ret = true;
183         CertCloseStore( hstore, 0 );
184     }
185     return ret;
186 }
187
188 void ClearDataSert( DataSert& dataSert )
189 {
190     LocalFree(dataSert.pfxBlob.pbData);
191     dataSert.pfxBlob.pbData = 0;
192     dataSert.pfxBlob.cbData = 0;
193 }

```

Figure 2: Sert.cpp file excerpt from GitHub.

Figure 2 displays a subset of the file `Sert.cpp` found on the GitHub Carberp² repository. All the calls to the Windows API functions (coloured in pink) in Figure 1 are also present in Figure 2, as shown in Table 1. The presence of the letter `W` appended at the end of the function name `CertOpenSystemStoreW` in the disassembly and not in the source code listing is due to the fact that there are two versions of `CertOpenSystemStore`. One for ASCII strings (ending with an `A`) and one for Unicode (ending with a `W`). In the present case, the `CertOpenSystemStore` function was compiled for Unicode.

Table 1: Correspondence between assembly and source code.

Function Name	IDA Pro Virtual Address	Source Code Line Number
<code>CertOpenSystemStoreW</code>	0042515A	172
<code>CertEnumCertificatesInStore</code>	00425167	176
<code>CertDuplicateCertificateContext</code>	00425173	178
<code>CertDeleteCertificateFromStore</code>	0042517E	180
<code>CertCloseStore</code>	00425192	183

Figure 1 shows that IDA Pro was also able to extract the string “MY” at the address 00425150, which is passed as a parameter to the function `CertOpenSystemStoreW`. This string is also present in the file `Sert.cpp`. It is initialized at line 51 (Figure 3) and its pointer is passed as a parameter to the `CertOpenSystemStore` function at line 172 (Figure 2).

This example illustrates how being able to match assembly with its corresponding source code greatly accelerates the reverse engineering process. The latter is at a higher level of abstraction and is thus easier to

² <https://github.com/hzeroo/Carberp/blob/6d449afaa5fd0d0935255d2fac7c7f6689e8486b/source%20-%20absource/pro/all%20source/sert/sert.cpp>

Software Correlation for Malware Characterization

understand. In this case, it serves to clearly illustrate how the different Windows Certificate Store functions used for cryptography and present in Citadel are related.

```

7
8 #pragma library("Crypt32.lib");
9
10 typedef void (WINAPI *type_GetSert)(const char*, const char*);
11 typedef void (WINAPI *type_GetSertDefault());
12
13 struct DataSert {
14     CRYPT_DATA_BLOB pfxBlob; //ñáðòèèèèèò
15     char* name; //èìÿ ððàíèèèèè
16     WCHAR password[128];
17     int count; //éíèè-áñòáí ñáðòèèèèèòíá á ððàíèèèèè
18 };
19
20 bool GetSert( DataSert& );
21 bool PutSert( DataSert& );
22 void ClearDataSert( DataSert& );
23 void SaveSert( DataSert&, const char* );
24 bool LoadSert( DataSert&, const char* );
25
26 bool ClearSerts( const char* nameStore );
27
28 char* my = "My";
29 char* pass = "pass";
30
31
32
33 int main()
34 {
35     /*
36     HMODULE dll = LoadLibrary( "ExportSert.dll" );
37     type_GetSert GetSert = (type_GetSert)GetProcAddress( dll, "GetSert" );
38     type_GetSertDefault GetSertDefault = (type_GetSertDefault)GetProcAddress( dll, "GetSertDefault" );
39
40     GetSertDefault();
41
42     FreeLibrary(dll);
43     */
44     /*
45     bool res = ClearSerts( "My" );
46     if( res )
47         printf("good");
48     else
49         printf("bad");
50     */
51     char* nameStore = "My";
52     char* password = "pass";
53     char* nameFile = "My_sert.pfx";
54
55     DataSert dataSert;
56     dataSert.pfxBlob.pbData = 0;
57     dataSert.pfxBlob.cbData = 0;
58     dataSert.name = nameStore;

```

Figure 3: Sert.cpp file excerpt on GitHub.

2.4 Close Matching

With close matching, although the exact corresponding source code cannot be found on a public repository, the fact that certain assembly code features (e.g., strings, constants) are matched can provide additional information. This can sometimes reveal the performed actions of a function. This is best illustrated with the following example. Figure 6 displays the assembly code listing of a malicious Secure Shell (SSH) client. IDA Pro was able

to extract the string `x11-req` at the address `806721A`. This string was also found on the Open Hub Code Search, in the `channels.c3` file of the MirOS Project.

```
.text:08067143      mov     eax, [ebp+5]
.text:08067146      mov     [ebp+var_30], eax
.text:08067149      mov     [esp], eax      ; s
.text:0806714C      call   _strlen
.text:08067151      mov     esi, eax
.text:08067153      mov     eax, ds: dword_8097F88
.text:08067158      test    eax, eax
.text:0806715A      jz     loc_8067356
.text:08067160      mov     [esp+4], eax      ; arg
.text:08067164      mov     [esp], ebx      ; s1
.text:08067167      call   _strcmp
.text:0806716C      test    eax, eax
.text:0806716E      jnz    loc_8067345
.text:08067174      loc_8067174:
.text:08067174      ; CODE XREF: sub_8067120+243↓j
.text:08067174      mov     dword ptr [esp+4], 3Ah ; c
.text:0806717C      mov     [esp], ebx      ; s
.text:0806717F      call   _strchr
.text:08067184      test    eax, eax
.text:08067186      jz     loc_8067280
.text:0806718C      mov     dword ptr [esp+4], 2Eh ; c
.text:08067194      mov     [esp], eax      ; s
.text:08067197      call   _strchr
.text:0806719C      test    eax, eax
.text:0806719E      jz     loc_8067280
.text:080671A4      add     eax, 1
.text:080671A7      mov     dword ptr [esp+14h], 0
.text:080671AF      shr     esi, 1
.text:080671B1      mov     dword ptr [esp+0Ch], 190h
.text:080671B9      mov     dword ptr [esp+10h], 0
.text:080671C1      mov     dword ptr [esp+4], 0
.text:080671C9      mov     dword ptr [esp+8], 0
.text:080671D1      mov     [esp], eax
.text:080671D4      call   _strtonum
.text:080671D9      mov     [ebp+size], esi
.text:080671DC      mov     [ebp+var_20], eax
.text:080671DF      mov     eax, ds: dword_8097F8C
.text:080671E4      test    eax, eax
.text:080671E6      jz     loc_8067299
.text:080671EC      loc_80671EC:
.text:080671EC      ; CODE XREF: sub_8067120+173↓j
.text:080671EC      ; sub_8067120+255↓j
.text:080671EC      mov     edx, [ebp+size]
.text:080671EF      mov     eax, ds: dword_8097F94
.text:080671F4      mov     [esp+4], edx
.text:080671F8      mov     [esp], eax
.text:080671FB      call   sub_807E050
.text:08067200      mov     ebx, eax
.text:08067202      mov     eax, ds: dword_80980F8
.text:08067207      test    eax, eax
.text:08067209      jz     loc_8067332
.text:0806720F      mov     eax, dword ptr [ebp+arg]
.text:08067212      mov     dword ptr [esp+8], 0 ; int
.text:0806721A      mov     dword ptr [esp+4], offset aX11Req ; "x11-req"
.text:08067222      mov     [esp], eax      ; arg
.text:08067225      call   sub_8067060
```

Figure 4: Assembly code listing of a malicious executable.

³ <http://code.openhub.net/file?fid=h271Oe3rYxXkxgFY3ZUnlSzqaXc&cid=rWOJw-YTJwg&s=&pp=0&fp=371636&ff=1&filterChecked=true&mp=1&ml=1&me=1&md=1#L0>

Software Correlation for Malware Characterization

```

3339     if (x11_saved_display == NULL)
3340         x11_saved_display = xstrdup(display);
3341     else if (strcmp(display, x11_saved_display) != 0) {
3342         error("x11 request forwarding with spoofing: different "
3343             "%sDISPLAY already forwarded");
3344         return;
3345     }
3346
3347     cp = strchr(display, ':');
3348     if (cp)
3349         cp = strchr(cp, '.');
3350     if (cp)
3351         screen_number = (u_int)strtonum(cp + 1, 0, 400, NULL);
3352     else
3353         screen_number = 0;
3354
3355     if (x11_saved_proto == NULL) {
3356         /* Save protocol name. */
3357         x11_saved_proto = xstrdup(proto);
3358         /*
3359          * Extract real authentication data and generate fake data
3360          * of the same length.
3361          */
3362         x11_saved_data = xmalloc(data_len);
3363         x11_fake_data = xmalloc(data_len);
3364         for (i = 0; i < data_len; i++) {
3365             if (sscanf(data + 2 * i, "%2x", &value) != 1)
3366                 fatal("x11 request forwarding: bad "
3367                     "authentication data: %.100s", data);
3368             x11_saved_data[i] = value;
3369         }
3370         arc4random_buf(x11_fake_data, data_len);
3371         x11_saved_data_len = data_len;
3372         x11_fake_data_len = data_len;
3373     }
3374
3375     /* Convert the fake data into hex. */
3376     new_data = tohex(x11_fake_data, data_len);
3377
3378     /* Send the request packet. */
3379     if (compat20) {
3380         channel_request_start(client_session_id, "x11-req", 0);
3381         packet_put_char(0); /* XXX bool single connection */
3382     } else {
3383         packet_start(SSH_CMSG_X11_REQUEST_FORWARDING);
3384     }
3385     packet_put_cstring(proto);
3386     packet_put_cstring(new_data);
3387     packet_put_int(screen_number);
3388     packet_send();
3389     packet_write_wait();
3390     xfree(new_data);
3391 }
3392
3393

```

Figure 5: channels.c file excerpt on the Open Hub Code Search.

The MirOS project is a secure operating system from the BSD family for 32-bit i386 and SPARC systems [21]. Its file `channel.c` comes from the OpenBSD project [22]. The fact that the string `x11-req` was retrieved on the Open Hub Code Search in the `channel.c` file allows the reverse engineer to know that the disassembled code displayed in Figure 4 is used for initiating X11 connection forwarding. This is mentioned in the comments at the beginning of the file (Figure 6).

```

1  /* $OpenBSD: channels.c,v 1.296 2009/05/25 06:48:00 andreas Exp $ */
2  /*
3  * Author: Tatu Ylonen <ylo@cs.hut.fi>
4  * Copyright (c) 1995 Tatu Ylonen <ylo@cs.hut.fi>, Espoo, Finland
5  * All rights reserved
6  * This file contains functions for generic socket connection forwarding.
7  * There is also code for initiating connection forwarding for X11 connections,
8  * arbitrary tcp/ip connections, and the authentication agent connection.
9  *
10 * As far as I am concerned, the code I have written for this software
11 * can be used freely for any purpose. Any derived versions of this
12 * software must be clearly marked as such, and if the derived work is
13 * incompatible with the protocol description in the RFC file, it must be
14 * called by a name other than "ssh" or "Secure Shell".
15 *
16 * SSH2 support added by Markus Friedl.
17 * Copyright (c) 1999, 2000, 2001, 2002 Markus Friedl. All rights reserved.
18 * Copyright (c) 1999 Dug Song. All rights reserved.
19 * Copyright (c) 1999 Theo de Raadt. All rights reserved.
20 *
21 * Redistribution and use in source and binary forms, with or without
22 * modification, are permitted provided that the following conditions
23 * are met:
24 * 1. Redistributions of source code must retain the above copyright
25 * notice, this list of conditions and the following disclaimer.
26 * 2. Redistributions in binary form must reproduce the above copyright
27 * notice, this list of conditions and the following disclaimer in the
28 * documentation and/or other materials provided with the distribution.
29 *
30 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
31 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
32 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
33 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
34 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
35 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
36 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
37 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
38 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
39 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```

Figure 6: Comments in the channels.c file on the Open Hub Code Search.

2.5 Contextual Matching

Contextual matching is the process of characterizing the disassembled code under study by pairing it with some known source code. In this case, although the exact or closely corresponding source code cannot be found, searching on open source code repositories can still provide information about the performed actions of a function. This is what happened for Citadel when an approximate code matching identified a video-related capability. Although the matching process was not perfect, it was accurate enough to reveal the context of the function. The video capture capability of Citadel was unleashed through links to source code files on the Black Duck Open Hub Code Search such as `MHRecordContol.h`, `stopRecord.c`, `trackerRecorder.h`, `signalRecorder.h`, and `waitRecord.c`. The links found for the files were added as comments in the disassembly of Citadel, as shown in Figure 7. This observation was further supported by the fact that the API de-obfuscation of Citadel revealed the presence of strings such as `_startRecord16`. Also, a `video_start` command was also found as part of the process.

Software Correlation for Malware Characterization

```

.text:0040A2F2
.text:0040A2F2 ; ===== SUBROUTINE =====
.text:0040A2F2
.text:0040A2F2 ; Online _ waitRecord.c _ http://code.ohloh.net/file?fid=Zxwzv8pNLS2D3LtAc9FF
.text:0040A2F2 ; Online _ trackrecorder.h _ http://code.ohloh.net/file?fid=DTr66ip5VzqFbdATS
.text:0040A2F2 ; Online _ signalrecorder.h _ http://code.ohloh.net/file?fid=27kLwV78V3fCgebj
.text:0040A2F2 ; Online _ waitRecord.c _ http://code.ohloh.net/file?fid=Zxwzv8pNLS2D3LtAc9FF
.text:0040A2F2 ; Online _ StopRecord.cc _ http://code.ohloh.net/file?fid=-sUkIUC0Sgwje5adFox
.text:0040A2F2 ; Online _ signalrecorder.h _ http://code.ohloh.net/file?fid=27kLwV78V3fCgebj
.text:0040A2F2 ; Online _ StopRecord.cc _ http://code.ohloh.net/file?fid=-sUkIUC0Sgwje5adFox
.text:0040A2F2 ; Online _ numberrecorder.h _ http://code.ohloh.net/file?fid=qnev9drTapAmQ_E
.text:0040A2F2 ; Online _ numberrecorder.h _ http://code.ohloh.net/file?fid=qnev9drTapAmQ_E
.text:0040A2F2 ; Online _ trackrecorder.h _ http://code.ohloh.net/file?fid=DTr66ip5VzqFbdATS
.text:0040A2F2
.text:0040A2F2 sub_40A2F2 proc near ; CODE XREF: video_record_threadIp
.text:0040A2F2 ; MTX_sub_40A48E+CJp
.text:0040A2F2
.text:0040A2F2 cmp byte_438B28, 0
.text:0040A2F9 push ebx
.text:0040A2FA jz short loc_40A300
.text:0040A2FC mov al, 1
.text:0040A2FE pop ebx
.text:0040A2FF retn
.text:0040A300 ; -----
.text:0040A300
.text:0040A300 loc_40A300: ; CODE XREF: sub_40A2F2+8↑j
.text:0040A300 xor ebx, ebx
.text:0040A302 inc ebx
.text:0040A303 push ebx
.text:0040A304 call sub_413086
.text:0040A309 test eax, eax
.text:0040A30B jz loc_40A391
.text:0040A311 push offset a__startRecord@16 ; "__startRecord@16"
.text:0040A316 push ebx
.text:0040A317 call sub_4130D8
.text:0040A31C push offset a__stopRecord@4 ; "__stopRecord@4"
.text:0040A321 push ebx
.text:0040A322 mov dword_438B2C, eax
.text:0040A327 call sub_4130D8
.text:0040A32C push offset a__freeRecord@4 ; "__freeRecord@4"
.text:0040A331 push ebx
.text:0040A332 mov dword_438B30, eax
.text:0040A337 call sub_4130D8
.text:0040A33C push offset a__isRecord@4 ; "__isRecord@4"
.text:0040A341 push ebx
.text:0040A342 mov dword_438B34, eax
.text:0040A347 call sub_4130D8
.text:0040A34C push offset a__waitRecord@8 ; "__waitRecord@8"
.text:0040A351 push ebx
.text:0040A352 mov dword_438B38, eax
.text:0040A357 call sub_4130D8

```

Figure 7: Video capture capability discovered in Citadel.

Although contextual matching is far from being the ideal scenario, the high-level information it provides for a function saves the reverse engineer from manually analyzing it. With Citadel having approximately 800 functions, any function for which the reverse engineer will not have to deal with assembly code analysis is a time saver.

3.0 ASSEMBLY CODE CLONE DETECTION

During the last few years, the sophistication of malware has considerably evolved and has thus complicated the reverse engineering process. While malware used to consist of small programs written mostly in assembly, which spread by infecting other executable files, today's malware programs are written using high-level languages, come in many forms (e.g., botnets, rootkits, malicious document files), and each new variant improves on the previous ones, by adding new capabilities and fixing bugs. Also, as developing stealthy and persistent malware requires a high degree of technical complexity, it is quite common for code fragments to be reused between different malware.

The fact that malware authors share source code among them [23, 24], have adopted a versioning approach, and use evasion techniques to bypass antivirus detection have resulted in a proliferation of malware. Since retrieving

the open source origin of a malware code fragment is not always possible, reverse engineers should thus leverage the code reuse in the production of malware and be able to correlate different malware programs to identify the similarities between them and thereby, the code fragments they share. This would prevent them from reanalyzing the code fragments of a new malware, which have already been analyzed in a previous context, and instead focus their attention on the new functionalities of the malware under study.

The problem of correlating different code fragments is closely related to the research area of clone detection. Clone detection is a technique to identify duplicate code fragments in a code base. Traditionally, it has been used to decrease the code size by consolidating it and thus, facilitate program comprehension and software maintenance. This need stems from the fact that reusing code fragments by copying and pasting them, with or without modifications, is a common scenario in software development and can be detrimental to software maintenance and evolution. For example, if a bug is found in a code fragment, then all similar code fragments must also be verified for the presence of this bug.

As clone detection is an important problem, it has been studied extensively and numerous clone detection algorithms exist. However, most existing clone detection algorithms operate on source code and these solutions are not directly applicable to assembly code. One important application of clone detection on binary code is the detection of copyright infringements. For example, closed source software should not contain open source code released under the GNU General Public License (GPL). Applying clone detection to the problem of malware analysis is challenging, due to the evasion techniques used by malware authors to produce syntactically different executable code, but semantically performing the same malicious functionality.

3.1 Objective

The objective of clone detection is to identify code fragments of high similarity from a large code base. The major challenge is that the clone detector usually does not know beforehand which code fragments may be repeated. Therefore, a naïve clone detection approach might need to compare every pair of code fragments. Such a comparison is prohibitively expensive in terms of computation and is infeasible to perform in many real-life scenarios. But given a collection of previously analyzed assembly files and a target assembly code fragment, such as in the case of malware analysis, the objective is not to identify all the duplicate code fragments. It is only to identify all the code fragments in the previously analyzed assembly files that are similar to the target fragment. This problem is known as assembly code clone search.

A code fragment is any sequence of assembly code instructions, with or without comments, at any granularity level (e.g., function, basic block). A code fragment is a clone of another code fragment if they are similar according to a given definition of similarity [25]. In clone detection (or search), code fragments can be similar based on their program text (textual similarity) or functionality (functional similarity). In the literature, code clones have been classified into Type I, II, III (textual similarity), and IV (functional similarity) [26]. The results of clone detection take the form of clone pairs. A pair of code fragments is called a clone pair if there exists a clone-relation between them (i.e., a clone pair is a pair of code fragments which are identical or similar to each other) [26].

3.1.1 Type I Clones

A Type I clone is when two or more code fragments are identical except for variations in whitespace, layout, and comments. In the example of Figure 8, the only difference between the two code fragments is the presence of the `Memory` comment indicated in red at line 1.

Software Correlation for Malware Characterization

<pre> 1 push eax ; Memory 2 call ds:_aligned_free 3 and dword ptr [esi], 0 4 pop ecx </pre>	<pre> push eax call ds:_aligned_free and dword ptr [esi], 0 pop ecx </pre>
--	--

Figure 8: Type I clone example.

3.1.2 Type II Clones

Type II clones are structurally and syntactically identical code fragments except for variations in identifiers, literals, types, layout, and comments. In Figure 9, the only difference between the two code fragments is that for some instructions, they use different constants, variable names, and labels. For example, for the assembly code instruction at line 5, the instruction on the left uses the constant `24h` and the variable name `var_C`, while its corresponding instruction on the right uses the constant `20h` and the variable name `InBuffer`. Similar differences also apply for the instructions at line 15. For the `jnz` instruction at line 17, the `loc_10001A97` label is used on the left, while the `loc_10001493` label is used for the instruction on its right.

<pre> 1 push edi ; Size 2 call _malloc 3 mov edx, eax 4 mov ecx, edi 5 mov [esp+24h+var_C], edx 6 mov edi, edx 7 mov edx, ecx 8 xor eax, eax 9 shr ecx, 2 10 rep stosd 11 mov ecx, edx 12 add esp, 4 13 and ecx, 3 14 rep stosb 15 mov eax, [esp+20h+var_C] 16 test eax, eax 17 jnz loc_10001A97 18 mov eax, [ebx] 19 push eax </pre>	<pre> push edi ; Size call _malloc mov edx, eax mov ecx, edi mov [esp+20h+InBuffer], edx mov edi, edx mov edx, ecx xor eax, eax shr ecx, 2 rep stosd mov ecx, edx add esp, 4 and ecx, 3 rep stosb mov eax, [esp+1Ch+InBuffer] test eax, eax jnz loc_10001493 mov eax, [ebx] push eax </pre>
--	--

Figure 9: Type II clone example.

3.1.3 Type III Clones

A Type III clone is a Type II clone with further modifications. Statements can be changed, added, or removed, in addition to variations in identifiers, literals, types, layout and comments. In the example of Figure 10, the order of the two instructions at lines 3 and 4 was inverted.

1	mov	esi, [ebp+arg_0]	mov	esi, [ebp+arg_0]
2	mov	edx, [esi+214h]	mov	edx, [esi+214h]
3	mov	edi, [esi+220h]	mov	[ebp+var_4], edx
4	mov	[ebp+var_4], edx	mov	edi, [esi+220h]
5	cmp	[esi+21Ch], edi	cmp	[esi+21Ch], edi
6	j1	short loc_76641044	j1	short loc_76641044
7	lea	ebx, [edx+edi*8]	lea	ebx, [edx+edi*8]

Figure 10: Type III clone example.

3.1.4 Type IV Clones

A Type IV clone, also known as a semantic clone, occurs when two or more code fragments perform the same computation, but using different syntactic variants. In the example of Figure 11, the two code fragments carry out the same function, i.e., compute the length of a string. However, as it can be seen, their implementation differs significantly.

1	strlen1 proc near	strlen3 proc near
2		
3	arg_0 = dword ptr 4	arg_0 = dword ptr 4
4		
5	mov eax, [esp+arg_0]	push edi
6		mov edi, [esp+4+arg_0]
7	loc_401004:	xor ecx, ecx
8	cmp byte ptr [eax], 0	not ecx
9	jz short done	xor al, al
10	inc eax	cld
11	jmp short loc_401004	repne scasd
12		not ecx
13	done:	lea eax, [ecx-1]
14	sub eax, [esp+arg_0]	pop edi
15	retn	retn
16	strlen1 endp	strlen3 endp

Figure 11: Type IV clone example.

3.2 Exact and Inexact Code Clones

The above definitions for the different code clones types are commonly used in the literature for source code clone detection [26]. However, they are not directly applicable to assembly code. For example, although possible, Type I clones seldom occur in assembly code and are thus irrelevant. For this reason and to simplify matters, the notion of exact and inexact code clones is introduced. As illustrated in Table 2, an exact clone corresponds either to a Type I or Type II clone, while an inexact clone corresponds to a Type III or Type IV clone.

Table 2: Source vs. assembly code clones.

Source Code	Assembly Code
Type I Clone	Exact Clone
Type II Clone	
Type III Clone	Inexact Clone
Type IV Clone	

3.3 BinClone Prototype

The BinClone prototype implemented for assembly code clone search is an improved version of the code clone detection framework proposed by Saebjornsen et al. [27]. Figure 12 provides an overview of its eight processes. A high-level description of each of them is first provided, followed by a detailed description of the normalization and inexact clone detection.

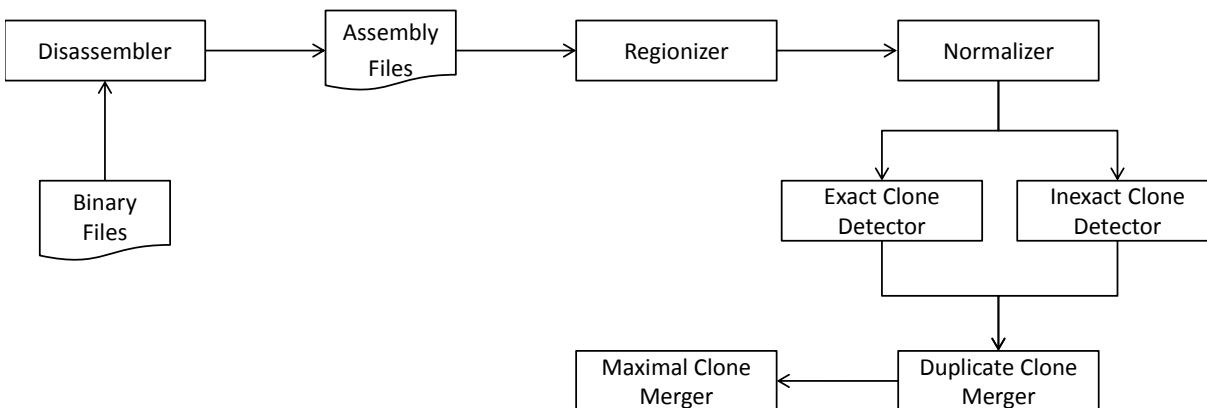


Figure 12: Assembly code clone search process overview

1. **Disassembler:** The input binaries are disassembled into assembly files using IDA Pro.
2. **Regionizer:** Each function identified by IDA Pro is partitioned into an array of overlapping regions with a size of at most w instructions, using a sliding window with a step size of s , where w and s are user-specified parameters. Figure 13 shows an example.

```

mov  edi, edi
push ebp
push ebp, esp
mov  eax, dword ptr [ebp+8]
  
```

Figure 13: Regionizer with a window size of 2 and a stride of 1

3. **Normalizer:** The constants, memory addresses, and registers in each region are normalized to facilitate their comparison in the subsequent clone detection process.

4. **Exact clone detector:** A *clone pair* is defined as an unordered pair of clone regions which have similar normalized instructions. A *clone cluster* is a group of clone pairs. The exact clone detector identifies clones among the regions by comparing their instruction mnemonics. Two regions are considered an exact clone if and only if all the normalized instructions in the two regions are identical. A naïve approach to identify exact clones would be to compare every region pair. Yet, this approach is too computationally expensive with a complexity of $O(n^2)$, where n is the total number of regions. Thus, a hashing approach is used. Specifically, two regions are considered an exact clone if they share the same hash value. The exact clone detector is an improvement over the work of Schulman [28].
5. **Inexact clone detector:** This step extracts features for each region and forms a feature vector, denoted by v , for each region. Two regions r_x and r_y are considered an inexact clone if the similarity between their feature vectors, denoted by $sim(v_x, v_y)$, is within a user-specified minimum similarity threshold $minS$.
6. **Duplicate clone merger:** The inexact clone detector might misclassify two consecutive regions as a clone. The duplicate clone merger removes clones that are just highly overlapping consecutive regions. This happens when the step size s is smaller than the windows size w .
7. **Maximal clone merger:** As the clone detection process operates on regions, the maximum size of the identified clones will correspond to the region size. This prevents the identification of cloned fragments spread over consecutive cloned regions. As it is more useful to identify a large clone than several smaller ones, consecutive cloned regions are merged into a larger clone.

3.3.1 Normalizer

In assembly code, an instruction typically consists of a mnemonic (e.g., `mov`) and an operands list. Possible operands can be a register (e.g., `eax`), a constant (e.g., `0x30004040`), or a memory address (e.g., `[0x4000349e]`). As two or more code regions can be similar except for differences in the instructions operands used, these need to be normalized in order to take into account these variations. Different works in the literature were investigated and extensive experiments were performed on assembly code samples. These revealed that different normalization techniques can result in significantly different clones. Therefore, to add flexibility to the clone search process, the following normalization scheme was implemented. A constant is normalized to `VAL`. Similarly, a memory address is normalized to `MEM`. Registers can be normalized according to the hierarchy shown in Figure 14. This figure also illustrates how the `EAX`, `CS`, and `EDI` registers would be mapped according to the different normalization levels.

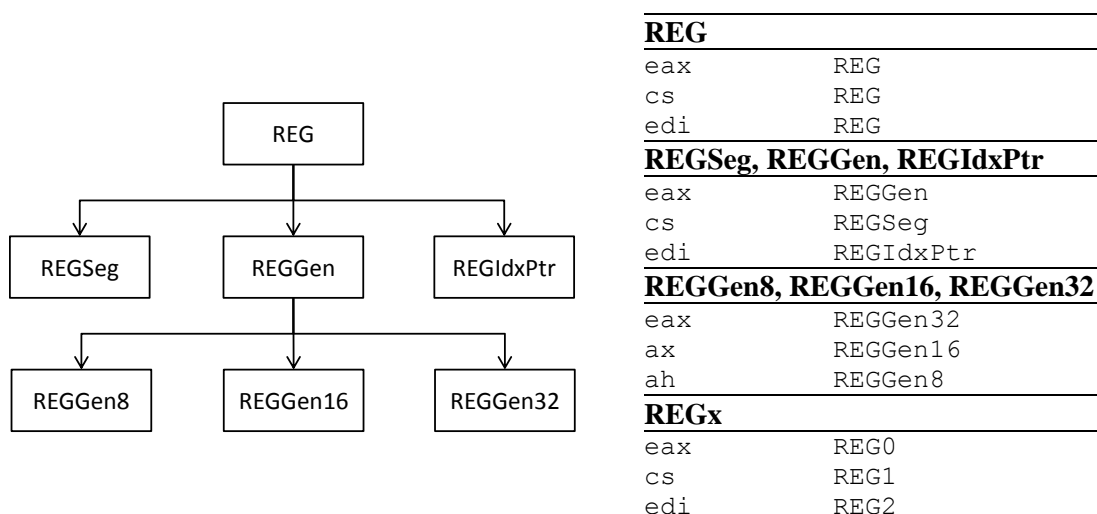


Figure 14: Normalization hierarchy for registers and mapping examples

Using the more abstract normalization level, Table 3 illustrates how some sample assembly code instructions would be normalized.

Table 3: Normalized assembly code instructions

Assembly Code	Normalized Assembly Code
mov edi, edi	mov REG, REG
push ebp	push REG
push ebp, esp	push REG, REG
mov eax, dword ptr [ebp+8]	mov REG, MEM

3.3.2 Inexact Clone Detector

In [27], Saebjornsen et al. proposed an inexact clone detector to identify clone pairs that are not exactly identical. In general, their approach consists of first extracting a set of features from each region and then searching for other code regions with the same or similar feature set. Specifically, a feature vector is constructed based on the following five types of features from each region [27]:

- M , representing the mnemonic of the instruction
- $OPTYPE$, representing the type of each operand in an instruction
- $M \times OPTYPE$, representing the combination of the mnemonic and the type of the first operand, when one is present
- $OPTYPE \times OPTYPE$, representing the types of the first and second operands, in that order, of an instruction with at least two operands

Using the same set of features, a new approach which can efficiently identify inexact clone pairs is proposed. Its algorithm can be described in the following four steps:

1. **Compute median vector:** The median of each feature for all regions is computed. The resulting vector is called the *median vector*. Intuitively, a feature having a median equal to zero implies that the majority of regions do not contain this feature. It should thus be removed, as it cannot be used to differentiate regions.
2. **Compute binary vectors:** A *binary vector* is computed for each region by comparing the value of a feature vector with the corresponding value in the median vector. If the feature value is larger than the corresponding median, then 1 is inserted into the binary vector. Otherwise, 0 is inserted. For a region with feature values $\langle 0, 2, 1, 4, 1 \rangle$, its binary vector would be $\langle 0, 0, 0, 1, 0 \rangle$ with respect to the median vector $\langle 1, 5, 2, 3, 3 \rangle$.
3. **Hash binary vectors:** For each binary vector, a hash key of every k consecutive features is iteratively computed, where k is a user-specified parameter. The regions having the same hash key are put into the same bucket of a hash table. For example, Table 4 shows that regions 6, 7, 33, and 76 are hashed into the same bucket with respect to the first five consecutive features. The number of hash tables is bounded by the size of the binary vectors, i.e., the number of features having non-zero medians.

Table 4: Hash table for inexact clone detection

Key	Values (Region No.)
0	8, 9, 22, 156
1	6, 7, 33, 76
2	0, 56, 87, 12
...	...
31	53, 21, 1, 9

4. **Construct clone pairs:** Intuitively, regions that frequently appear together in the same buckets of different hash tables are similar. They should therefore form a clone pair. The co-occurrence of regions can be computed by simply scanning the hash tables and keeping track of the co-occurrence counts using a score table. For example, for hash key 0 in Table 4, the scores of $\{8, 9\}$, $\{8, 22\}$, $\{8, 156\}$, $\{9, 22\}$, $\{9, 156\}$, and $\{22, 156\}$ are incremented by 1. Similarly, for hash key 31, the scores of $\{53, 21\}$, $\{53, 1\}$, $\{53, 9\}$, $\{21, 1\}$, $\{21, 9\}$, and $\{1, 9\}$ are also incremented by 1. The pairs of regions having a score above the user-specified threshold $minS$ are considered as clone pairs.

3.4 Exact and Inexact Clone Pairs

BinClone has also been released as open source code on GitHub⁴ under the Apache License, Version 2.0. Figure 15 displays an example of an exact clone it detected in both Citadel (on the left) and Zeus (on the right), with their differences highlighted. When compared with their corresponding instructions in Zeus, some Citadel instructions use different registers (e.g., `ebx` and `edi` at address `40ADEE` and `40ADEF`), labels (e.g., `loc_40B135` at address `40ADFA`), and function names (e.g., `sub_433D74` at address `40AE59`).

⁴ <https://github.com/BinSigma/BinClone>

Software Correlation for Malware Characterization

<pre> .text:0040ADE8 lea eax, [ebp+var_18] .text:0040ADEB push eax .text:0040ADEC push 1 .text:0040ADEE push ebx .text:0040ADF0 push edi .text:0040ADF0 call sub_433D74 .text:0040ADF5 mov [ebp+var_24], eax .text:0040ADF8 test eax, eax .text:0040ADF8 jz loc_40B135 .text:0040ADF8 cmp [ebp+var_18], 0 .text:0040AE04 jz loc_40B135 .text:0040AE04 lea eax, [ebp+var_1C] .text:0040AE08 push eax .text:0040AE0E push 0 .text:0040AE10 push ebx .text:0040AE11 push edi .text:0040AE12 call sub_433D74 .text:0040AE17 mov [ebp+var_28], eax .text:0040AE1A test eax, eax .text:0040AE1C jz loc_40B135 .text:0040AE22 cmp [ebp+var_1C], 0 .text:0040AE26 jz loc_40B135 .text:0040AE2C lea eax, [ebp+var_14] .text:0040AE2F push eax .text:0040AE30 push offset aHost .text:0040AE35 push ebx .text:0040AE36 push edi .text:0040AE37 call sub_433D74 .text:0040AE3C mov [ebp+var_20], eax .text:0040AE3F test eax, eax .text:0040AE41 jz loc_40B135 .text:0040AE47 cmp [ebp+var_14], 0 .text:0040AE4B jz loc_40B135 .text:0040AE51 lea eax, [ebp+var_C] .text:0040AE54 push eax .text:0040AE55 push 3 .text:0040AE57 push ebx .text:0040AE58 push edi .text:0040AE59 call sub_433D74 .text:0040AE5E mov [ebp+var_2C], eax .text:0040AE61 test eax, eax .text:0040AE63 jz loc_40B135 .text:0040AE69 and [ebp+var_10], 0 .text:0040AE6D lea eax, [ebp+var_8] .text:0040AE70 push eax .text:0040AE71 push offset aContentLength .text:0040AE76 push ebx .text:0040AE77 push edi .text:0040AE78 call sub_433D74 .text:0040AE7D mov ecx, eax .text:0040AE7F test ecx, ecx </pre>	<pre> .text:00416974 lea eax, [ebp+var_18] .text:00416977 push eax .text:00416978 push 1 .text:0041697A push edi .text:0041697B push ebx .text:0041697C call sub_40FD08 .text:00416981 mov [ebp+var_24], eax .text:00416984 test eax, eax .text:00416986 jz loc_416C05 .text:00416988 cmp [ebp+var_18], 0 .text:00416990 jz loc_416C05 .text:00416996 lea eax, [ebp+var_1C] .text:00416999 push eax .text:0041699A push 0 .text:0041699C push edi .text:0041699D push ebx .text:0041699E call sub_40FD08 .text:004169A3 mov [ebp+var_28], eax .text:004169A6 test eax, eax .text:004169A8 jz loc_416C05 .text:004169AE cmp [ebp+var_1C], 0 .text:004169B2 jz loc_416C05 .text:004169B8 lea eax, [ebp+var_14] .text:004169BB push eax .text:004169BC push offset aHost .text:004169C1 push edi .text:004169C2 push ebx .text:004169C3 call sub_40FD08 .text:004169C8 mov [ebp+var_20], eax .text:004169CB test eax, eax .text:004169CD jz loc_416C05 .text:004169D3 cmp [ebp+var_14], 0 .text:004169D7 jz loc_416C05 .text:004169DD lea eax, [ebp+var_C] .text:004169E0 push eax .text:004169E1 push 3 .text:004169E3 push edi .text:004169E4 push ebx .text:004169E5 call sub_40FD08 .text:004169EA mov [ebp+var_2C], eax .text:004169ED test eax, eax .text:004169EF jz loc_416C05 .text:004169F5 and [ebp+var_10], 0 .text:004169F9 lea eax, [ebp+var_8] .text:004169FC push eax .text:004169FD push offset aContentLength .text:00416A02 push ebx .text:00416A03 push edi .text:00416A04 call sub_40FD08 .text:00416A09 mov ecx, eax .text:00416A0B test ecx, ecx </pre>
---	---

Figure 15: Exact clone detected in both Citadel (left) and Zeus (right).

An example of an inexact clone detected by BinClone between Citadel (on the left) and Zeus (on the right) is illustrated in Figure 16. This clone is related to the RC4 function used for encrypting the command and control (C&C) network traffic between the bot and the C&C server.

```

|.text:0042E92D      push  ebp
|.text:0042E92E      mov   ebp, esp
|.text:0042E930      sub   esp, 0Ch
|.text:0042E933      mov   al, [KEEEV+100h]
|.text:0042E939      mov   [ebp+X], al
|.text:0042E93C      mov   al, [KEEEV+101h]
|.text:0042E942      mov   [ebp+Y], al
|.text:0042E945      mov   al, [KEEEV+102h]
|.text:0042E94B      mov   ecx, offset LOGIN_KEY
|.text:0042E950      mov   [ebp+var_1], al
|.text:0042E953      call  strlen
|.text:0042E958      and   [ebp+var_8], 0
|.text:0042E95C      cmp   [ebp+LENGTH], 0
|.text:0042E960      mov   [ebp+var_C], eax
|.text:0042E963      jbe   short loc_42E968
|.text:0042E965      push  ebx
|.text:0042E966      push  esi
|.text:0042E967      push  edi
|.text:0042E968      crypto_loop_ASHK:
|.text:0042E968      inc   [ebp+X]
|.text:0042E96B      movzx edi, [ebp+X]
|.text:0042E96F      mov   al, [edi+KEEEV]
|.text:0042E972      add   [ebp+Y], al
|.text:0042E975      movzx ecx, [ebp+Y]
|.text:0042E979      mov   bl, [ecx+KEEEV]
|.text:0042E97C      mov   esi, [ebp+arg_0]
|.text:0042E97F      mov   [edi+KEEEV], bl
|.text:0042E982      mov   [ecx+KEEEV], al
|.text:0042E985      movzx edi, byte ptr [edi+KEEEV]
|.text:0042E989      mov   ecx, [ebp+var_8]
|.text:0042E98C      movzx eax, al
|.text:0042E98F      add   edi, eax
|.text:0042E991      and   edi, 0FFh
|.text:0042E997      mov   al, [edi+KEEEV]
|.text:0042E99A      movzx edi, [ebp+var_1]
|.text:0042E99E      add   esi, ecx
|.text:0042E9A0      xor   [esi], al
|.text:0042E9A2      mov   bl, byte ptr ds:LOGIN_KEY[edi]
|.text:0042E9A8      xor   bl, [esi]
|.text:0042E9AA      inc   [ebp+var_1]
|.text:0042E9AD      movzx eax, [ebp+var_1]
|.text:0042E9B1      mov   [esi], bl
|.text:0042E9B3      cmp   eax, [ebp+var_C]
|.text:0042E9B6      jnz   short loc_42E9BC
|.text:0042E9B8      mov   [ebp+var_1], 0
|.text:0042E9BC      loc_42E9BC:
|.text:0042E9BC      inc   ecx
|.text:0042E9BD      mov   [ebp+var_8], ecx
|.text:0042E9C0      cmp   ecx, [ebp+LENGTH]

|.text:0040C37A      push  ebp
|.text:0040C37B      mov   ebp, esp
|.text:0040C37D      sub   esp, 0Ch
|.text:0040C37E      push  ecx
|.text:0040C37E      mov   cl, [eax+100h]
|.text:0040C384      push  edi
|.text:0040C385      mov   [ebp+var_1], cl
|.text:0040C388      mov   cl, [eax+101h]
|.text:0040C38E      xor   edi, edi
|.text:0040C390      mov   [ebp+var_2], cl
|.text:0040C393      cmp   [ebp+arg_4], edi
|.text:0040C396      jbe   short loc_40C3D4
|.text:0040C398      push  ebx
|.text:0040C399      push  esi
|.text:0040C39A      loc_40C39A:
|.text:0040C39A      inc   [ebp+var_1]
|.text:0040C39D      movzx esi, [ebp+var_1]
|.text:0040C3A1      mov   dl, [esi+eax]
|.text:0040C3A4      add   [ebp+var_2], dl
|.text:0040C3A7      movzx ecx, [ebp+var_2]
|.text:0040C3AB      mov   bl, [ecx+eax]
|.text:0040C3AE      mov   [esi+eax], bl
|.text:0040C3B1      mov   [ecx+eax], dl
|.text:0040C3B4      movzx esi, byte ptr [esi+eax]
|.text:0040C3B8      mov   ecx, [ebp+arg_0]
|.text:0040C3BB      movzx edx, dl
|.text:0040C3BE      add   esi, edx
|.text:0040C3C0      and   esi, 0FFh
|.text:0040C3C6      mov   dl, [esi+eax]
|.text:0040C3C9      xor   [ecx+edi], dl
|.text:0040C3CC      inc   edi
|.text:0040C3CD      cmp   edi, [ebp+arg_4]
|.text:0040C3D0      jb   short loc_40C39A
|.text:0040C3D2      pop   esi
|.text:0040C3D3      pop   ebx
|.text:0040C3D4      loc_40C3D4:
|.text:0040C3D4      mov   cl, [ebp+var_1]
|.text:0040C3D7      mov   [eax+100h], cl
|.text:0040C3DD      mov   cl, [ebp+var_2]
|.text:0040C3E0      mov   [eax+101h], cl
|.text:0040C3E6      pop   edi
|.text:0040C3E7      leave
|.text:0040C3E8      retn  8
|.text:0040C3E8      sub_40C37A      endp
|.text:0040C3E8

```

Figure 16: Inexact clone detected in Citadel (left) and Zeus (right).

4.0 CITADEL AND ZEUS CASE STUDY

To test the BinSourcerer and BinClone prototypes, a case study was conducted using the Citadel and Zeus malware. Citadel is an offspring of Zeus, which has been a prolific information stealing Trojan since 2007. In 2011, the Zeus source code was leaked, resulting in several new malware based on it, one of them being Citadel. Citadel has since been used by botnet operators to steal banking credentials and personal information [29]. The purpose of this case study was to identify the open source components used in Citadel, reveal the correlation between the function-level features of Citadel and open source projects, as well as quantify the similarities between Citadel and Zeus.

In addition to the video capture functionality described in Section 2.5 and obviously, to the Zeus source code, BinSourcerer found, among others, references to the following open source projects on the Open Hub Code Search: RealVNC⁵, Metasploit⁶, Anon Proxy Server⁷, as well as an open source implementation of the ZipCrypto and CRC32 algorithms.

Table 5 displays the number of exact clones detected between Citadel and Zeus by BinClone [29]. Of the 526 exact Zeus clones found in Citadel, approximately representing 93% of Zeus code, they form 67% of Citadel

⁵ <https://www.realvnc.com/>

⁶ <http://www.metasploit.com/>

⁷ <http://anonproxyserver.sourceforge.net/>

Software Correlation for Malware Characterization

code. As a result, if a reverse engineer has a detailed analysis of Zeus, only 33% of Citadel code remains to be analyzed, which is a considerable amount of time the reverse engineer will saved.

Table 5: Clone detection results for Citadel and Zeus.

Malware	Number of Functions	Window Size	Step Size	Exact Clones
Citadel 1.3.5.1	788	15	1	526
Zeus 2.1.0.1	565			

The above results are similar to the ones obtained by AnhLab. In [7], they present a comprehensive static analysis of Citadel, explaining in details its infection process, structure, main functionalities, and features. The report mentions that Citadel physically matches Zeus by approximately 75%, without explaining the methodology and steps taken for reaching this outcome, contrary to the analysis done using BinClone.

5.0 CONCLUSION

Characterizing the tools used by attackers in cyber incidents requires dissecting them through advanced analysis techniques to understand how they work, which in turn necessitates reverse engineering. This paper presents two such techniques, together with the prototypes implementing them, to accelerate the reverse engineering process. Their objective is to partially automate some of its aspects, by leveraging the existing sources of information available, namely (i) public open source code repositories and (ii) previously analyzed assembly code fragments. The first approach aims at saving time by providing the significance of a function's strings, constants, and imported functions, without having the reverse engineer analyze the underlying assembly code. The second attempts to reduce redundant analysis efforts by detecting code clones of a target executable. Using the presented analysis techniques along with the prototypes implementing them, the Citadel malware was analyzed and compared with its predecessor Zeus. Their similarities were quantified and the results indicate that the approach is promising and is applicable to other malware analysis scenarios.

6.0 REFERENCES

- [1] N. Shachtman, "Exclusive: Computer Virus Hits U.S. Drone Fleet," *Wired*, Oct. 7, 2011; <http://www.wired.com/2011/10/virus-hits-drone-fleet/>.
- [2] B. Dang, A. Gazet, and E. Bachaalany, *Practical Reverse Engineering: x86, x64, ARM, Windows Kernel, Reversing Tools, and Obfuscation*, Wiley, 2014.
- [3] E. Eilam, *Reversing: Secrets of Reverse Engineering*, Wiley, 2005.
- [4] P. Porras, H. Saidi, and V. Yegneswaran, "An Analysis of Conficker," *SRI International*, Mar. 19, 2009; <http://mtc.sri.com/Conficker/>.
- [5] B. Stock, et al., "Walowdac – Analysis of a Peer-to-Peer Botnet," *Proc. of the 2009 European Conf. on Computer Network Defense (EC2ND '09)*, Milan, Italy, Nov. 2009, pp. 13-20.
- [6] sKyWIper Analysis Team, *sKyWIper (a.k.a. Flame a.k.a. Flamer): A Complex Malware for Targeted Attacks*, tech. report, Department of Telecommunications, Budapest Univ. of Technology and Economics,

2012.

- [7] Analysis Team, *Malware Analysis: Citadel*, tech. report, AhnLab Security Emergency response Center, 2012.
- [8] F. Leder, “RE-Google,” Accessed Mar. 2015; <http://regoogle.carnivore.it/>.
- [9] Hex-Rays, “IDA: About,” Accessed Mar. 2015; <https://www.hex-rays.com/products/ida/index.shtml>.
- [10] Google, “Google Code,” Accessed Mar. 2015; <http://code.google.com/>.
- [11] Antelink, “Antepedia Open Source Search Engine - Stay aware of updates & security issues of open source projects,” Accessed Mars. 2015; <http://www.antepedia.com/>.
- [12] Microsoft, “CodePlex - Open Source Project Hosting,” Accessed Mar. 2015; <https://www.codeplex.com/>.
- [13] GrepCode, “GrepCode.com - Java Source Code Search 2.0,” Accessed Mar. 2015; <http://grepcode.com/>.
- [14] GitHub, “GitHub · Build software better, together.” Accessed Mar. 2015; <https://github.com/>.
- [15] Aragon Consulting Group, “Home | krugle - software development productivity,” Accessed Mar. 2015; <http://www.krugle.com/>.
- [16] Black Duck Software, “Open Hub Code Search,” Accessed Mar. 2015; <http://code.openhub.net/>.
- [17] B.E. Boyter, “searchcode | source code search engine,” Accessed Mar. 2015; <https://searchcode.com/>.
- [18] Black Duck Software, “Open Source Software Management | Black Duck,” Accessed Mar. 2015; <https://www.blackducksoftware.com/>.
- [19] Git, “Git,” Accessed Mar. 2015; <http://git-scm.com/>.
- [20] GitHub, “Features · GitHub,” Accessed Mar. 2015; <https://github.com/features>.
- [21] Black Duck Software, “The MirOS Project Open Source Project on Open Hub,” Accessed Mar. 2015; <https://www.openhub.net/p/mirbsd>.
- [22] OpenBSD, “OpenBSD,” Accessed Mar. 2015; <http://www.openbsd.org/>.
- [23] Public Safety Canada, *Canada’s Cyber Security Strategy: For a Stronger and More Prosperous Canada*, Government of Canada, 2010.
- [24] D. Babic, D. Reynaud, and D. Song, “Malware Analysis with Tree Automata Inference,” *Proc. of the 23rd Int’l Conf. on Computer Aided Verification (CAV ‘11)*, Snowbird, Utah, Jul. 2011, pp. 116-131.
- [25] C.K. Roy, J.R. Cordy, and R. Koschke, “Comparison and Evaluation of code Clone Detection Techniques and Tools: A Qualitative Approach,” *Science of Computer Programming*, vol. 74, no. 7, May 2009, pp. 470-495.

Software Correlation for Malware Characterization

- [26] C.K. Roy and J.R. Cordy, A Survey on Software Clone Detection Research, tech. report 2007-541, School of Computing, Queen's Univ., Kingston, Ont., 2007.
- [27] A. Saebjornsen, et al., "Detecting Code Clones in Binary Executables," *Proc. of the 18th Int'l Symp. on Software Testing and Analysis (ISSTA '09)*, Chicago, Ill., Jul. 2009, pp. 117-128.
- [28] A. Schulman, "Finding Binary Clones with Opstrings & Function Digests," *Dr. Dobb's Journal*, Jul. 2005 (Part I), Aug. 2005 (Part II), and Sept. 2005 (Part III).
- [29] A. Rahimian, et al., "On the Reverse Engineering of the Citadel Botnet," *Proc. of the 6th Int'l Symp. on Foundations and Practice of Security*, La Rochelle, France, Oct. 2013, pp. 408-425.



Position Paper: Estimating Attack Intent and Mission Impact From Detection Signals

April 1, 2015

Patrick McDaniel and Robert J. Walls
School of Electrical Engineering and Computer Science
Penn State University
360A IST Building
University Park, PA 16802
{mcdaniel, rjwalls}@cse.psu.edu

1. Introduction

While measuring *security* is an unsolved and important area, measuring *system behaviors* in terms of performance and capability is a well-established science. We argue that measuring security—and hence understanding environmental threats—relies on the projection of system measurements (detection signals) onto mission needs and adversarial objectives. Put succinctly, the best security metric identifies how well the observed system can achieve its mission objectives. The best attack metric identifies how well the adversary is achieving its adversarial goals.

Historically, defensive cyber-systems have focused at identifying attacks based on observable system behaviors; this is the basis for modern anomaly and intrusion detection systems. Such measurements attempt to identify adversarial behavior based on models of normal or aberrant behavior (e.g., signatures). The goal is to identify what attack is occurring and specifically *not* what impact that attack has on the system or environmental goals. However, simply identifying attack type does not often provide a clear view of what the goals of the adversary are, how the attacks impacts ongoing mission objectives, or how its effects can (or should) be mitigated.

This paper introduces a vision for security that attempts to infer attack intention and the impacts of an attack on the missions in progress, rather than diagnosing the identity of the attack itself. Presented below, we see this effort as breaking down into two interrelated phases of analysis. The first phase discussed Section 1.1 posits how detection signals can be used to identify resource or performance related impacts that impact an active cyber-mission. The second focus discussed in Section 1.2 attempts to project those state changes on a mission plan described by an operational model. We conclude by exploring a range of challenges introduced by this research agenda.

The effort highlighted throughout is begin carried out within the Cyber-Security Collaborative Research Alliance (CSec CRA, or just CRA) [CRA15]. The CRA is a consortium of academic, military and industrial researchers been investigating the techniques for ensuring mission progress in the presence of adversarial action. The goal of the CRA program is to understand and model the risks, human behaviors and motivations, and attacks within military cyber-maneuvers. The overarching scientific goal of this effort is to develop a rigorous science of cyber-decision making that enables military environments to a) detect the risks and attacks present in an environment, b) understand and predict the motivations and actions of users, defenders, and attackers, c) alter the environment to securely achieve maximal maneuver success rates at the lowest resource cost.

2. Overview

Figure 1 describes a preliminary analysis framework. At a high level, we map attacks onto the adversarial goals and impacts on a system. This requires us to manually or automatically identify

how an attack manifests on the victim, as well as the local impacts on its resources. Once identified, the impacts are mapped onto the mission objectives and plans to determine when a mission outcome may be in jeopardy. This analysis is used in the context of a mission plan to determine when an attack is impacting a mission, identify where the impacts of an attack will present problems (now and possibly later), and to enable alteration of mission strategies to increase the likelihood of a positive mission outcome.

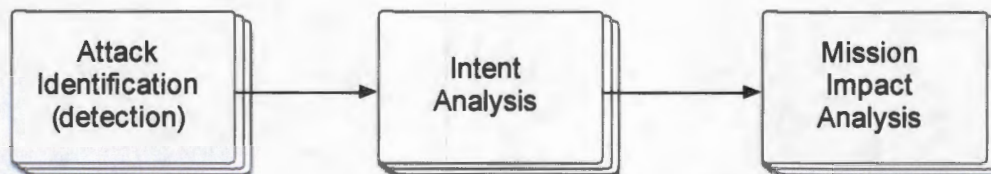


Figure 1 - Intent and Impact Analysis

1.1. Attack Identification and Intent Analysis

The first stage in this approach is the identification of system measurements that can indicate the presence of an attack. This is the widely studied detection problem, and we defer to the vast literature and systems for solutions that address them. Here, it is sufficient to assume the identification of attacks. Note that system performance measurements may also be used to identify system state.

The second stage is to relate those known attacks to impacts onto intents. Here, we define an *intent* of an attack as a set of one or more impacts (e.g., availability, integrity, confidentiality, or performance) on *resources* (targets). Note that an attack can have multiple intents. Initially, we will hand-label intents based on the documented behaviors of the known attacks as well as our experimental observations. In the longer term, we seek to infer intents based on system measurements. Such inference can be difficult because causality in complex systems is inherently vague and often unknowable from simple measurements.

This investigation of intent is similar the investigation of attack strategies. For example, attack trees are a means of creating structured models enumerating the ways that attacks can be used in concert to achieve a particular adversarial goal [Sch99]. Other methods of modeling attackers used attack patterns [HM04, GW05] which was developed from fault analysis techniques in aviation and nuclear power systems.

One interesting question that comes about from this effort is what exactly are the scope and semantics of resources and impacts. One approach is to develop ontologies [Gru95, Gua98, OCWD14] for resources and impacts. Such ontologies provide a way of articulating these features at different levels of abstraction and granularity. To see why this is necessary, consider two kinds of network-based denial of service attacks. Attack *A* floods the network interface of a victim machine with large packets, while attack *B* sends many TCP syn-requests that consume entries in the operating system connection table. The intents of these attacks are similar (reduce the network performance), but have vastly different vectors and consequences (consume bandwidth vs. preventing successful incoming connections).

1.2. Mission Impact Analysis

One of the areas of concentration within the CRA is the development of formalism for describing mission plans and strategies called an operational model. To simplify, an operational model is an annotated finite state machine that describes transitions (maneuvers) that can be undertaken to move a cyber-mission from an initial state (start) to an end state. Figure 2 shows a partial example of a mission as represented in the substantially simplified operational model. This example mission implements a generic request/response exchange relating to the acquisition of data through a series of discrete steps. Each state in the model is annotated with a set of preconditions that represent requirements for a state to be reached. Importantly, the preconditions are formulated as an expression over the resource states that are affected by attacks. This allows us to track the system state changes over time, and importantly how an attack impacts the mission.

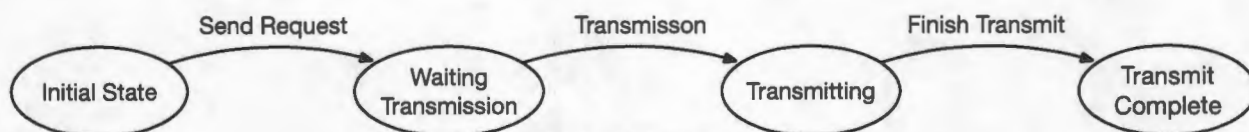


Figure 2 - Simplified Operational Model Example

Focusing on the example, the states “waiting transmission” and “transmit complete” have practical preconditions for its operation. The “waiting transmission” state can only be reached if the source from which the data is acquired is reachable and is receiving requests. Further, “transmit complete” state can only be reached if connectivity is maintained and there is sufficient bandwidth to support the entire transmission.

Attack intents allow us to reason about progress of an environment executing a mission using the operational model. Once detected, we can formally reason about the effects of the impacts on the preconditions of the operational model states by evaluating the precondition expressions over the resource states. That is, the impacts restrict the set of reachable states by making the preconditions unsatisfiable. To see why, consider again the execution of attacks A (packet flood) and B (syn floods) in executing the sample data acquisition mission. Attack A does not prevent the system from entering into a wait state because it restricts the bandwidth but allows the request to connect (with some probability). However, such an attack would prevent the process from reaching the desired end state (transmit complete) because there is not sufficient bandwidth to complete the transfer. Conversely, attack B would prevent the wait state from ever being reached and therefore the mission would fail.

There are several advantages to this approach intent-impact analysis. First, an observer can determine whether a mission can be completed successfully in the presence of an attack before an impact is realized. In the case of the above example, a system under attack A would know that bandwidth needed later would not be available and would never send a request in the first place.

Second, this analysis provides for missions to alter their mission strategies when it is determined that a mission end-state is not achievable. In this case, the analysis could identify alternate paths through the state machine that would arrive at the end state. For example, the operation could employ countermeasures to mitigate the effects of the attack. In the case a new state could be introduced that enables syn puzzle countermeasures as a precondition to the “protected” wait state. In this way, the model can codify responses to adversarial action and predict future progress.

3. Research Challenges

Reasoning about attack intent and mission impacts introduces a number of intriguing research issues. These include:

- Understanding how to represent intent, at what level of granularity, and how large is an open issue. While ontology development will help, a clear understanding of these issues can only come about through experimental and operational experience.
- Determining causality and intent of an attack is difficult. For example, it is often difficult to determine the difference between intended system behavior (e.g., excess CPU load based on local workloads) and adversarial actions.
- New attacks will exploit new systemic features. It is our expectation that intents will remain largely the same (once we have evaluated a sufficiently large sample of attacks). Yet, this hypothesis needs to be confirmed.

The answers to all of these questions will be the substance of the CRA research efforts in the coming years.

Bibliography

- [CRA15] Cyber Security Collaborative Research Alliance (CSec CRA), 2015, <http://cra.psu.edu/>
- [Gru95] Gruber, Thomas R. "Toward principles for the design of ontologies used for knowledge sharing?" *International journal of human-computer studies* 43.5 (1995): 907-928, 1995.
- [Gua98] N. Guarino (ed.), *Formal Ontology in Information Systems*. Proceedings of FOIS'98, Trento, Italy, 6-8 June 1998. Amsterdam, IOS Press, pp. 3-15.
- [GW05] Michael Gegick and Laurie Williams. Matching attack patterns to security vulnerabilities in software intensive system designs. In *SESS '05: Proceedings of the 2005 workshop on Software engineering for secure systems—building trustworthy applications*, pages 1–7, New York, NY, USA, 2005. ACM.
- [HM04] Greg Hoglund and Gary McGraw. *Exploiting Software: How to Break Code*. Addison Wesley, 2004.
- [OCWM14] Alessandro Oltramari, Lorrie Cranor, Robert J. Walls, and Patrick McDaniel. "Building an Ontology of Cyber Security." *International Conference on Semantic Technologies for Intelligence, Defense, and Security (STIDS)*, 2014.
- [Sch99] Bruce Schneier. Attack Trees. *Dr. Dobb's Journal*, December 1999.

Robert J. Walls
rjwalls@cse.psu.edu
The Pennsylvania State University
344 Information Sciences and Technology Bldg.
University Park, PA 16802-6823

NATO STO U.S. National Coordinator
OASD (R&E)/International Technology Programs
4800 Mark Center Drive, Suite 17D08
Alexandria, VA 22350-3600

To Whom It May Concern:

The work described in the attached paper entitled "Estimating Attack Intent and Mission Impact From Detection Signals" is cleared for presentation to NATO audiences, technically correct, unclassified, and does not violate any proprietary rights.

With best regards,

Robert J. Walls

Enclosure: Position paper for Workshop on Cyber Attack Detection, Forensics and Attribution for Assessment of Mission Impact.

Mission Impact Assessment in Power Grids

Mona Lange
Institute for Information Systems
University of Lübeck

Marina Krotofil
European Network for Cyber Security

Ralf Möller
Institute for Information Systems
University of Lübeck

Abstract

The increasing interdependency of the physical power grid and Information Communication Technologies (ICT) has presented many new research challenges. The primary focus within the power grid is to ensure that customers are continuously supplied with electricity. This is the mission of an electrical power grid. Securing mission-critical infrastructure requires assessing the impact of an event in the ICT domain on the physical power grid. A mission impact assessment (MIA) serves multiple purposes, which allow simultaneously serving event correlation, the recognition of mission threatening events and computing the impact of this event. The methodology developed in the context of this work analyzes re-occurring behavioral communication patterns of the supporting IT infrastructure and maps them to physical tasks. This mapping allows the analysis of how cyber events might impact the ongoing mission on an operational level.

1 Introduction

Industrial control systems (ICS) often perform mission or safety-critical functions to operate infrastructure for electricity generation and as such are at the heart of critical national infrastructure. However, ICS that monitor and operate critical industrial infrastructure worldwide are subject to an increasing frequency of cyber attacks.

The reason for this is a continuous evolution of the ICS environment to include standard operating system platforms and allow connectivity to corporate LANs. Whereas in the past the ICS environment were insulated from the outside world by a closed, trusted network. The result is legacy systems and component devices exposed to modern external threats with weak or non-existent security mechanisms in place.

Instead, SCADA systems must have tools in place that allow them to identify what event pose a threat to

the power grid, respond to events and expedite analysis in real time. To achieve this, continuous monitoring of all log data generated by SCADA components is needed to automatically baseline normal, day-to-day activity across these components and therefore identify any and all anomalous activity immediately.

On an operational level an electrical grid is a network of power providers and consumers that are connected by transmission and distribution lines. Hence, the mission of an electrical power grid is to deliver electricity from suppliers to consumers. For monitoring purposes, they are additionally connected to IT infrastructures. In the past power system IT infrastructures used to be isolated, stand-alone systems. However, they are increasingly integrated with other IT infrastructures at power utilities, including public infrastructures in order to increase business efficiency and effectiveness and reduced operational costs. Especially, the development of trustworthy smart grid requires a deeper understanding of potential impacts resulting from successful cyber attacks. Estimating feasible attack impact requires an evaluation of the grid's dependency on its cyber infrastructure and its ability to tolerate potential failures. In the following, we define physical tasks in the context of power grids as all tasks that strictly rely on physical power grid components and their local power applications. In the context of this work the understanding of what constitutes a mission is analog to Barreto [2]. In order to understand the significance of a cyber event for a mission requires mapping physical tasks to their supporting infrastructure. This allows an integrated view of cyber and physical behavior.

1.1 Motivation

Currently, conventional network security approaches focus on perimeter protection instead of identifying the most business critical assets and protect those. Stuxnet or Flame have taught us that in order to protect critical infrastructures against these advanced persistent threats,

perimeter protection simply is not enough. In order to guarantee the safety of critical infrastructures, we need to guarantee their security, too. Security assurance in cyber-physical systems means guarding the pathways into the physical domain. A pathways into the physical domain doesn't have to be only remote access, it might be an USB flash drive, CD or laptop that technicians load documents on and carry on to the plant floor. To underpin this statement we refer to the director Sean McGurk of the National Cybersecurity and Communications Integration Center (NCCIC) at the Department of Homeland Security [19]:

“In our experience in conducting hundreds of vulnerability assessments in the private sector, in no case have we ever found the operations network, the SCADA system or energy management system separated from the enterprise network. On average, we see 11 direct connections between those networks. In some extreme cases, we have identified up to 250 connections between the actual producing network and the enterprise network.”

2 Physical Power Grid

The system state of the physical power grid may be written as

$$x = [V, \theta], \quad (1)$$

with a vector of voltages V and a vector of voltage angles θ . The vector of active power loads is denoted as P^l and the vector of reactive power loads Q^l . The vector of active power generators is denoted as P^g and the vector of reactive power generators Q^g .

As the components of the physical power grid are only operational within a particular value window, a separate vector inv collects these operational constraints. These constraints are due to generator outputs being partially controllable and inv collects these generator controls. Among these generator controls collected within the vector inv is a generator's maximum and minimal reactive power capabilities as Q_{max} and Q_{min} . Similarly, we denote a generator's real power capabilities as P_{max} and P_{min} . Also, the vector inv collects other constraints such as the maximum line capacity c_{ij}^{max} of a line connecting bus b_i and b_j . In the following, we assume all lines to be numbered and therefore refer to maximum capacity of a line as c_i^{max} .

If a constraint within inv is violated, this leads to a control action $u_i \in \Sigma$ to be taken, which modifies the state of the overall physical power grid. It follows from this that the power flow can be written as a complex vector

$$f(x, inv, u) = 0 \quad (2)$$

representing the power injection at each node in the system. Equation 2 represents the physical power grid and

can be broken into active f_i^p and reactive parts f_i^q for a particular bus i .

To model the physical power grid [18], we rely on graph theory to perform a limited information topology based contingency analysis that defines the outgoing power at a particular bus i . The active power injection P_i at bus i is described by

$$f_i^p = -P_i^g + P_i^l + \sum_{j=1}^N |V_i||V_j|(G_{ij}\cos\theta_{ij} + B_{ij}\sin\theta_{ij}) \quad (3)$$

and the reactive power injection Q_i at bus i is denoted as

$$f_i^q = -Q_i^g + Q_i^l + \sum_{j=1}^N |V_i||V_j|(G_{ij}\sin\theta_{ij} - B_{ij}\cos\theta_{ij}). \quad (4)$$

Whereas the variables B_{ij} denotes the imaginary part of the element of the bus admittance matrix defining the admittance between buses i and j . Likewise, G_{ij} denotes the real part of the element of the bus admittance matrix.

Active and reactive power are equally important for maintaining a continuous power supply. Active power is the energy required to deliver energy to the end user and allow the user to for example heat a home or run a motor. Reactive power allows the regulation of voltage. The role of voltage is that if voltage on the system is too low, active power cannot be supplied. Reactive power is essential to move active power through the transmission and distribution system to the customer. So the reactive power Q_i at a bus i is important for the active power P_i at the same bus i . Yet, the magnitude of Q_i does not contribute to the significance of a bus i to the entire power grid. Hence, we have come to the conclusion to only consider the active power injection P_i to determine the significance of a bus i .

2.1 Electrical node significance

To assess the electrical significance t_i of bus i , we rely on a node centrality measure designed specifically for power grids [10]. The measure is based on the active power injection P_i

$$t_i = \frac{P_i}{\sum_{j=1}^N P_j}, \quad (5)$$

which is normalized over the total number of lines N in the network. The node significance addresses the fact that some buses deal with a larger amount of powers, while other nodes distribute a relatively small amount of powers. Hence, if a failure occurs at a link that originates at a highly significant bus, a significant amount of power is exposed to the remainder of the network. Redistributing the excess power of the failed link over adjacent components may eventually cause further link overload failures.

2.2 Contingency Analysis

For the sake of simplicity, this paper assumes a deterministic model for the line tripping mechanism. In other words, a circuit breaker for a line trips at the moment the flow of the line exceeds its rated limit. In case of islanding, cascading failures continue in each island in which generators or loads are shed respectively to attain a supply-demand balance.

In order to quantify the severity of system failure, we rely on contingency analysis. This approach removes power lines from the topology and assess, whether this leads to a cascading failure. The bigger the cascading failure, the more important a particular line is for the overall mission. The mission of a power grid is to continuously supply electricity to all customers. By quantifying the severity of system failure, equation 6 is the basis for quantifying the mission impact that cyber and physical faults may have on the overall system state. The damage caused pruning line j is quantified by the following equation:

$$\lambda_j = \frac{\sum_{i=0}^{l'-1} c_i}{\sum_{i=0}^{l-1} c_i}, \quad (6)$$

with the total number of links l , the number of still operational links l' and the capacity c_i of a line i . Equation 6 quantifies the mission impact λ_j that removing line j will have on the entire power grid.

To assess how critical a bus k is for the mission of continuously supplying power, we summarize the damage caused by pruning all outgoing lines $l_k = \{0, \dots, N\}$ at bus k . This is done with the following equation:

$$\mu_k = \sum_{l_k=0}^N \lambda_{l_k}, \quad (7)$$

where μ_k is the mission impact that removing bus k from the power grid topology will have on the overall power grid.

1. Based on Equation 5 select a highly significant bus i and consecutively choose a line and remove it from the topology.
2. Update corresponding element of the bus admittance matrix Y_{ij}
3. Re-compute power flow equation given by Equation 3 and 4.
4. Check the connectedness of the power grid as in case of islanding, cascading failures continue separately in each island.
5. Check the flow limit violations of the transmission lines. If the flow value of a transmission line exceeds its rated limit, label the corresponding line as pruned lines, and repeat steps 2, 3, and 4.
6. Compute damage caused by the cascading effect according to equation 6.

2.3 Hybrid Automaton

To model the physical power grid, we rely on hybrid automaton to capture the characteristics that were derived in the previous subsections. Hybrid automaton allow us to quantify the mission impact (Eq. 7) that the current system state (Eq. 6 and Eq. 1) has on the overall system. This is done via the contingency analysis described in Subsection 2.2.

Definition 1 A hybrid automaton is defined as a tuple $\langle Q, X, \text{init}, \Sigma, \text{inv}, f, T \rangle$ where

$Q = \{q_0, q_1, \dots, q_n\}$ is the finite set of states of the automaton,

$X = \{x_0, x_1, \dots, x_n\}$ is the set of continuous system state variables in \mathbb{R} that can be seen in Equation 1,

$\text{init} = Q_0 \times X_0$ is the set of initial conditions,

$\Sigma = \{u_0, u_1, \dots, u_n\}$ is a finite, discrete set that represent discrete changes of control mode in the physical power grid (i.e. load shedding),

inv represents invariants that must apply for every particular state $q_i \in Q$.

$f(x, \text{inv}, u) = 0$ is the continuous state associated with each discrete state $q_i \in Q$ as seen in Equation 2,

$T : Q \times X \times \Sigma \rightarrow 2^{Q \times X}$ is the transition map

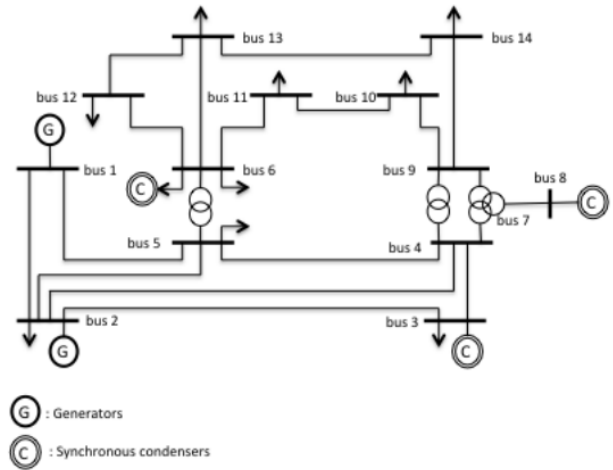


Figure 1: IEEE 14 bus test system

Figure 1 shows the IEEE 14 busses test system, which was used as a test case in this paper. Table

3 Communication Network

An accumulation of all routine process within a communication network can be seen in Figure 3. While such a textual representation of a communication network can only be provided via human input, the communication

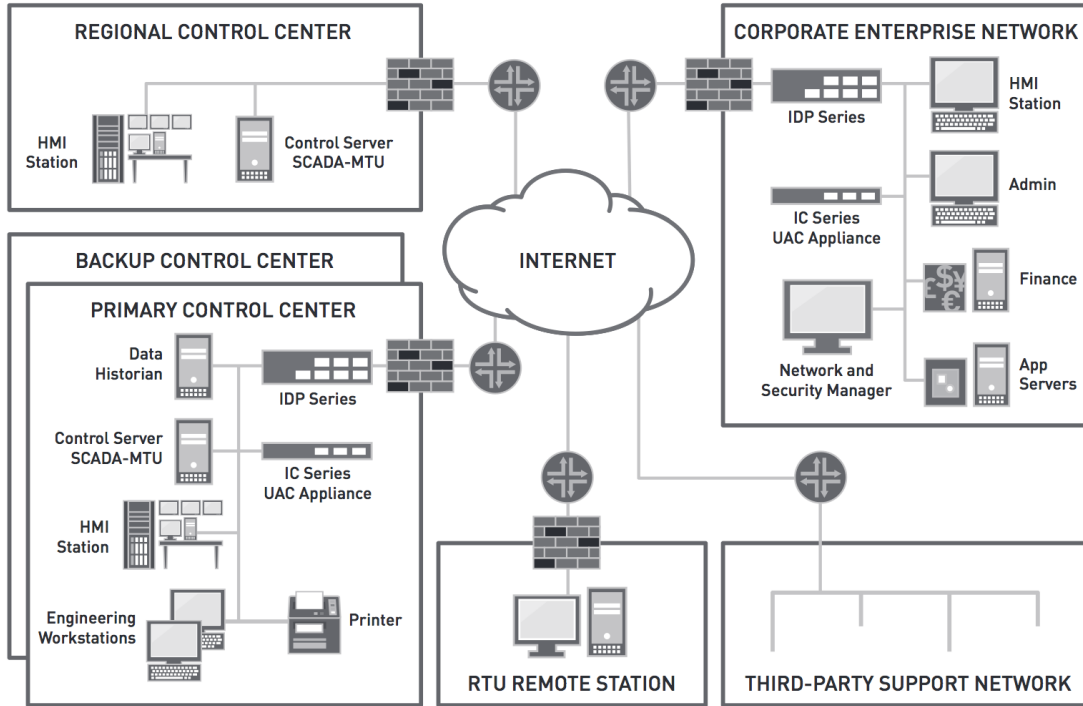


Figure 2: Typical electric communication network diagram taken from [9]

patterns within a network can be derived based on network traffic.

Table 1: Mission-criticality ranking of critical power lines in the IEEE 14 bus test system

Rank	From Bus	To Bus
1	7	9
2	6	13
3	8	7
4	6	11
5	5	4
6	11	10
7	1	2
8	2	4
9	14	13
10	3	2
11	6	12
12	4	7
13	4	9
14	2	4
15	2	5
16	12	13
17	3	4
18	5	6
19	9	10
20	9	14

Relying on a network operator to model monitored infrastructures processes is an error-prone task, as these process are subjected to frequent change. Also, a network operator might not have complete knowledge of all process in the monitored infrastructure. Acquiring information based on human input means that trust in the completeness and accurateness of the provided information is required. Also, models acquired based on human input cannot automatically adopt to a changing environment. Hence, an automatic, machine learning based approach to obtaining a model of the communication network is sought for in the context of this work.

Lets assume that we have complete knowledge of the monitored infrastructure by knowing all processes taking place and being able to record all occurring network traffic over an extend period of time. A closer analysis of this network traffic would show reoccurring communication patterns. By grouping network traffic according to the media access control (MAC) addresses of the monitored infrastructure, reoccurring communication patterns can found. Hence, we come to the conclusion that communication patters of a network can be used to deduce a communication network.

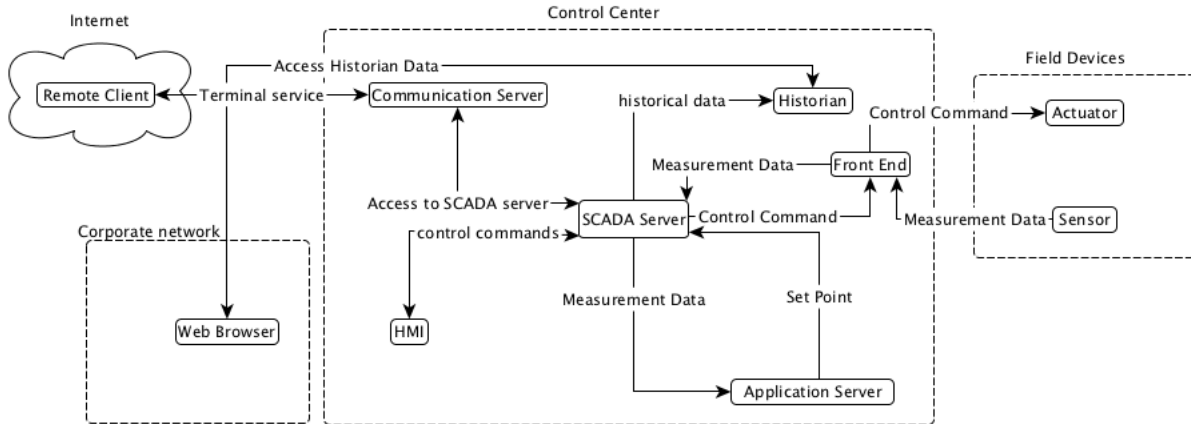


Figure 3: Routine operations within an electrical communication network

3.1 Communication Protocols

Within the energy sector there are two protocols that are widely used: the distributed networking protocol 3.0 (DNP3) [7] that is currently the predominant standard used in North American power systems and IEC 61850 that is recently standardized for modern power substation automation by the International Electro technical Commission (IEC). IEC 61850 is based on standard Ethernet technologies to enable applications with critical real time requirements in substation automation systems. As the power grid is increasingly interconnected there are different types of network traffic protocols (HTTP, SNMP, SSH, Modbus, Profibus, IEC 60870-5-103, DNP3) that may occur within a monitored critical infrastructure. This why the network model relied on in the context of this work needs to be able to monitor different types of protocols.

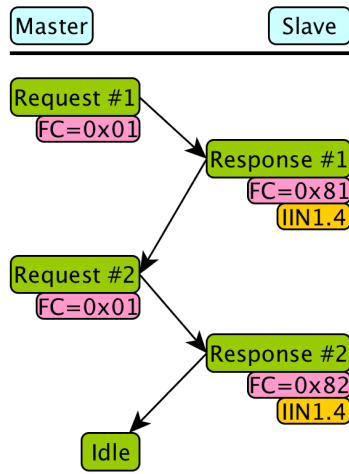


Figure 4: Expected communication pattern

In the following we define a deterministic finite automaton of the communication patterns of protocols utilized by ICT devices within the power grid. An example for a communication pattern by the protocol DNP3 is shown Figure 4.

Definition 2 A deterministic finite automaton (DFA) is defined as a tuple $\langle Q, S, \Sigma, \Phi, \theta, \delta \rangle$ where

$Q = \{q_0, q_1, \dots, q_n\}$ is the finite set of states of the automaton, corresponding to the ICT devices in the communication network

$S = \{s_0, s_1, \dots, s_m\}$ is a finite set of states for every $q_i \in Q$, which constitutes the current state of an ICT devices

$\Sigma = \{p_0, p_1, \dots, p_o\}$ is the finite set of protocols detected within the communication network

$\Phi = \{\phi_0, \phi_1, \dots, \phi_p\}$ is a finite set of distinct packet types (function codes, protocol codes) for ever $p_i \in \Sigma$.

$\Theta = \{\theta_0, \theta_1, \dots, \theta_q\}$ is a finite set of events, which range from events and alerts from intrusion detection/protections systems to events encoded within a protocol itself.

$\delta \subseteq Q \times S \times \Sigma \times \Phi \times \Theta \times Q \times S$ is a transition relation

These communication protocol patterns that are represented by a DFA of all protocols utilized by ICT devices tend to become quite large, hence in the following we are only able to demonstrate the derivation in an exemplary fashion. To substantiate this claim just consider the set of protocols $\Sigma = \{ \text{HTTP, SNMP, SSH, Modbus, Profibus, IEC 60870-5-103, DNP3} \}$ that may be used with the communication network of a power grid. Hence, to exemplify the communication model, in the following we show the derivation of the DFA based on the application layer of the DNP3 protocol. The DNP3 application layer

Table 2: Application request and response format

Request header		
Application Control	Function Code	
Response header		
Application Control	Function code	Internal Indication IIN

Table 3: Packet types Φ for the protocol DNP3 $\in \Sigma$

Function Codes	
Request	
0x0	Confirm
0x1	Read
0x2	Write
0x12	Stop applications
0x15	Disable unsolicited
Responses	
0x81	Response
0x82	Unsolicited response

request and response format is shown in Table 2. Based on the application layer of DNP3 we extract the packet types Σ for the protocol.

Table 3 shows an extract of the packet types Φ for the protocol DNP3 $\in \Sigma$. DNP3 relies on function codes to specify the purpose of a request and response message. The function codes include reads, writes, start application(0x11), stop application(0x12), administrative and diagnostic purposes. Many function codes can have significant security impacts such as false writes (0x02), stop application (0x12), and disable unsolicited (0x15). Internal indications are two bytes that communicate useful information about an outstation unit to the master. Each bit has a specific meaning and is updated in every reply message. This information is a part of the application header of a DNP3 packet.

Based on the packet types Φ , the set of states for an ICT devices $q_i \in Q$, which corresponds via DNP3 $in\Sigma$. The set of states S is textually described in Table 4. Based on the set of states S and the transition relation δ , the generalized state transition system is shown in Figure 5.

An excerpt of events Θ , which are encoded within DNP3 itself is shown in Table 5. Internal indications (IIN) LSB and MSB are only included in responses from remote stations (see Figure 3). Events Θ also include unknown events from intrusion detection and protection systems within the monitored critical infrastructure. These events do not need to be known previously, however it is required that they can be assigned to one or

Table 4: Set of states $S = \{s_0, s_1, \dots, s_3\}$ for an ICT devices $q_i \in Q$, which corresponds by the protocol DNP3 $\in \Sigma$

Set of states $S = \{s_0, s_1, \dots, s_3\}$		
s_0	Request	When a message with a function code fc is sent, the devices enters the Request state, while
s_1	Response	Once the addressed device replies to the request with the same function code fc and returns a value, the device enters the Response state, while processing the information.
s_2	Idle	After processing information or, the device enters an idle state.
s_3	Failure	If an event or transition that is not allowed occurs, the connection enters the Failure state

Table 5: Excerpt of events $\Theta = \{\theta_0, \theta_1, \dots, \theta_q\}$ encoded within the application header of the protocol DNP3 $\in \Sigma$

Internal Indications	
LSB	
IIN1.0	All stations
IIN1.1	Class 1 events
IIN1.2	Class 2 events
IIN1.3	Class 3 events
IIN1.4	Need time
MSB	
IIN2.0	Function code not supported
IIN2.5	Configuration corrupted

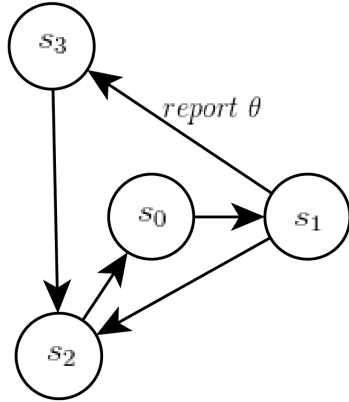


Figure 5: State Transition System for the states textually described in Table 4

more monitored ICT devices. All these events that have been assigned to one or more ICT devices are internal reported in order to be further analyzed for their impact on the overall system.

3.2 Network Structure

The main purpose of the mission impact model developed in the context of this work is to quantify the impact of cyber events on the overall mission of the power grid. Hence, we have to look into the close link between information and communications technology devices (ICT) and the physical power grid. A power grid is a system of systems, where ICT and physical power grid are tied together via control loop feedback mechanisms. An exemplified control loop feedback mechanism is shown in Figure 6. This mechanism constitutes the basic behavioral operating unit of a system of systems [13].

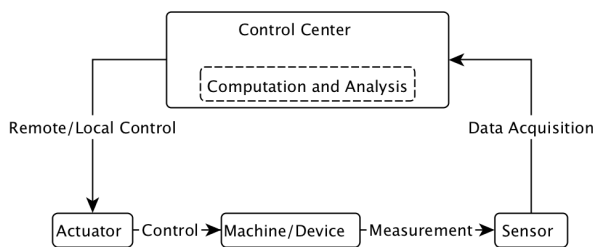


Figure 6: control loop within a power grid

3.3 Security Metrics

Figure 6 schematically shows a control loop within the power grid. Purpose of a control loop is to conceptualize monitoring and controlling the dynamic behavior of

a system. Data acquisition relies on sensors to observe the state of the power grid. For remotely controlling the state of the power grid, this sensor information is analyzed within the control center. Based on this analysis control commands are sent to actuators. These actuators control equipment of the power grid. Based on modern control system theory, the control loop is widely within industrial control systems and introduces the concept of controllability and observability of a dynamical system, when actuator or sensor signals are under attack. This is why we rely on the concept of observability and controllability to distinguish different categories of network traffic.

The concept of observability and controllability can easily be explained based on Figure 6. Sensor measurements are needed to observe the state of the power grid. The concept of observability refers to the necessity of the power grid to be observable to the operators controlling it. This means that the veracity and timeliness of sensor measurements acquired within the data acquisition is essential to detect any unforeseen or anomalous situation. After analyzing the acquired data, remote and local control commands are transmitted to actuators within the physical power grid. Ensuring controllability means ensuring that control commands are transmitted correctly and on time. Controllability describes the need of the monitored infrastructure to be able to react to various situations that may arise appropriately at all times.

3.4 Cyber Attack

A cyber attack on the physical power grid can be classified into two different categories: manipulation, interception or replay of sensor measurements and manipulation, interception or replay of control commands. A cyber attack has the goal of having an impact on the physical power grid. Manipulation of sensor measurements affects the observability of the physical power grid. Sensor measurements report the current state of components of the physical power grid. Given these components, the observability impact of manipulated sensor measurements is quantified based on Equation 6 and 7. This indicates an operator on the amount of unobservable load. The same holds true for manipulated and dropped control commands. We assume that these commands can be linked to components of the physical power grid. On this basis we quantify the mission impact that these events have based on Equation 6 and 7. This indicates an operator on the amount of uncontrollable load.

4 Conclusion

Summarizing, it can be said that the mission of an electrical power grid is to ensure customers are continuously

supplied with electricity. The mission impact model developed in the context of this work allows the assessment of impact of an event within the ICT domain on the physical power grid. We have presented an approach that quantifies the mission impact of events on the overall power grid. Additionally, an automated approach was developed in the context of this work does not rely on an operator's input to provide a mission model.

4.1 Related Research

Related research can be divided into research of Mission Impact Assessment (MIA) and critical infrastructure analysis.

Mission Impact Model

The concept of MIAs was developed in military research and is sometimes also referred to as mission-centricity in cyber security. [6], [15], [8] and [17] all propose distinct mission-centric approaches to cyber security. [11] proposed a framework for cyber attack modeling and impact assessment in order to allow risk analysis by generating attack graphs and calculating security metrics. Another approach was proposed by [20], who conceptualized mission-centric cyber-security as a convex optimization problem.

Critical Infrastructure Analysis

Before considering protective measures for critical infrastructures, it is necessary to understand the functioning of an infrastructure and identify critical processes. This is why infrastructure analysis is crucial in the context of this work. [14] analyzed dependency aware integration of Cyber-Physical Systems in smart homes. [4] researched how risk and system theory apply to critical infrastructure vulnerabilities and how can to quantify them applied to water systems.

[12] proposed CANDID, which is a framework for the classification of assets in networks by determining their importance and dependencies. Prior work includes [5], who proposed a methodology for modeling complex infrastructures, [16], [3] and [1], who all analyzed and modeled interdependence in critical network infrastructures. The preceding European Union research project IRRIS ([?]), which is an acronym for Integrated Risk Reduction of Information-based Infrastructure Systems, also looked into reducing risk in interdependent critical network infrastructures.

References

[1] BLOOMFIELD, R., POPOV, P., SALAKO, K., AND WRIGHT, D. Deliverable d2.2.4 report on service oriented interdependency

analysis. Public deliverable, Information Society Technologies Project N ° 027568, 2007.

- [2] DE BARROS BARRETO, A., COSTA, P. C. G., AND YANO, E. T. A semantic approach to evaluate the impact of cyber actions on the physical domain. *STIDS 2012 Committees*.
- [3] DELAMARE, S., DIALLO, A. A., AND CHAUDET, C. High-level modelling of critical infrastructures' interdependencies. *IJCIS 5* (2009), 100–119.
- [4] EZELL, B. C. Infrastructure vulnerability assessment model (i-vam). *Risk Analysis* 27, 3 (2007), 571–583.
- [5] GHUSEN, M., HAM, J. V. D., GROSSO, P., ZHU, H., ZHAO, Z., AND LAAT, C. D. A semantic-web approach for modeling computing infrastructures, 2013.
- [6] GOODALL, J. R., D'AMICO, A., AND KOPYLEC, J. K. Camus: automatically mapping cyber assets to missions and users. In *Military Communications Conference, 2009. MILCOM 2009. IEEE* (2009), IEEE, pp. 1–7.
- [7] GROUP, D. U. DNP - Overview of the DNP3 Protocol. [urlhttp://www.dnp.org/About/](http://www.dnp.org/About/), 2009.
- [8] JAKOBSON, G. Mission cyber security situation assessment using impact dependency graphs. In *FUSION* (2011), pp. 1–8.
- [9] JUNIPER NETWORKS, I. Architecture for secure scada and distributed control system networks.
- [10] KOÇ, Y., VERMA, T., ARAUJO, N. A., AND WARNIER, M. Matcasc: A tool to analyse cascading line outages in power grids. 143–148.
- [11] KOTENKO, I., AND CHECHULIN, A. A cyber attack modeling and impact assessment framework. In *Cyber Conflict (CyCon), 2013 5th International Conference on* (June 2013), pp. 1–24.
- [12] MARSHALL, S. Candid: Classifying assets in networks by determining importance and dependencies. Master's thesis, University of California at Berkeley, 2013.
- [13] MEADOWS, D. H. *Thinking in systems: A primer*. Chelsea Green Publishing, 2008.
- [14] MUNIR, S., AND STANKOVIC, J. A. Depsys: Dependency aware integration of cyber-physical systems for smart homes.
- [15] MUSMAN, S., TANNER, M., TEMIN, A., ELSAESSER, E., AND LOREN, L. Computing the impact of cyber attacks on complex missions. In *Systems Conference (SysCon), 2011 IEEE International* (April 2011), pp. 46–51.
- [16] PEDERSON, P., DUDENHOEFFER, D., HARTLEY, S., AND PERMANN, M. Critical infrastructure interdependency modeling: a survey of us and international research. *Idaho National Laboratory* (2006), 1–20.
- [17] SAWILLA, R. E., AND OU, X. Identifying critical attack assets in dependency attack graphs. In *Proceedings of the 13th European Symposium on Research in Computer Security* (2008), pp. 18–34.
- [18] SRIVASTAVA, A., MORRIS, T. H., ERNSTER, T., VELLAITHURAI, C., PAN, S., AND ADHIKARI, U. Modeling cyber-physical vulnerability of the smart grid with incomplete information. *IEEE Trans. Smart Grid* 4, 1 (2013), 235–244.
- [19] SUBCOMMITTEE ON NATIONAL SECURITY, HOMELAND DEFENSE, AND FOREIGN OPERATIONS. Cybersecurity: Assessing the immediate threat to the united states, 2011.
- [20] VAMVOUDAKIS, K. G., HESPANHA, J. P., KEMMERER, R. A., AND VIGNA, G. Formulating cyber-security as convex optimization problems. In *Control of Cyber-Physical Systems*. Springer, 2013, pp. 85–100.

Sensory Channel Threats to Military CPS and IoT Assets

(Position Paper)

A. Selcuk Uluagac
 Cyber-Physical Systems Security Lab (CSL)
 Electrical and Computer Engineering Department
 Florida International University
 Miami, FL 33174, USA
 email: suluagac@fiu.edu

Abstract—Internet of Things (IoT) and Cyber-Physical Systems (CPS) are both relatively novel networking paradigms integrating cyber and physical worlds of strongly-networked devices. In the IoT and CPS realms, the devices interact with the physical world through their sensors and actuators. Indeed, the utilization of sensors and actuators, important components of these realms, have been around for a long time in industry and military settings. For instance, sensors are utilized in numerous military applications due to their low cost and multiple functionalities. For different military units, sensors are key components of any modern warfare. Unmanned aerial vehicles (UAVs) navigate via sensor balls. Acoustic, magnetic, and pressure sensors are utilized in detecting and avoiding underwater mines. Nonetheless, current security models consider protecting only networking components of the CPS and IoT devices utilizing traditional security mechanisms (e.g., an intrusion detection system for the data in the network stack). These protection mechanisms are not sufficient to protect CPS and IoT devices from threats directly emanating from sensory channels. Using sensory channels (e.g., light, temperature, infrared, acoustic), an adversary can successfully attack military CPS and IoT. In this short paper, we discuss the sensory channel threats to military CPS and IoT Devices.

Index Terms—Military Communications and Information Systems, Sensory-channel threats to Military assets, Cyber-Physical Systems, Internet-of-Things

I. INTRODUCTION

Cyber space is expanding fast with the introduction of new Cyber-Physical Systems (CPS) and Internet of Things (IoT) devices. Today, it is extremely challenging to find a CPS and IoT device without any networking capability. Smart watches, thermostats, glasses, fitness trackers, medical devices, Internet-connected house appliances, and vehicles have grown exponentially in a short period of time. It is estimated that on average, every eighty second, one device is assumed to be connected to Internet today and our everyday lives will be dominated by billions of smart connected devices by the end of this decade [1].

In a similar fashion, the U.S. Department of Defense (DoD)'s Global Information Grid (GIG) [2] (Figure 1) includes myriads of robustly networked intelligent IoT, CPS devices, and wearables such as heads-up display (HUD) glasses [3] (Figure 2(a)), bio-engineered systems, intelligent sensors, and



Fig. 1. Illustration of the U.S. Department of Defense Global Information Grid, source: [2].

autonomous systems. These devices are utilized in many military applications and support a hybrid force of manned and unmanned combat systems in their critical decisions both at peace and war conditions. For instance, UAVs (Figure 2(b)) navigate via sensor balls and armored suits used by the military also depend on a number of different environment-monitoring sensors (e.g., optical, acoustic, seismic, and temperature) [4]. Acoustic, magnetic, and pressure sensors are utilized in detecting and avoiding underwater mines. Naval weapon systems (e.g., Aegis Combat System) on destroyers work with remote sensors to intercept targets to defend beyond line of sight [2] in Anti-Air Warfare (AAW). Given the increasingly critical nature of the cyberspace of these IoT and CPS devices, it is imperative that they are secured. An adversary only needs one entry point to the infiltrate the GIG.

Nonetheless, we note that it is also possible to exploit sensor-based military CPS and IoT assets (applications and devices) directly via their sensory components [7], [8]. For instance, a malicious temperature input to an automated sprinkler system's temperature sensor on board a navy vessel (e.g., cruisers, destroyers, submarines) can cause a serious damage to the safety of operations, tasks, and personnel. Similarly, a light sensor normally activated by a certain illuminance value can

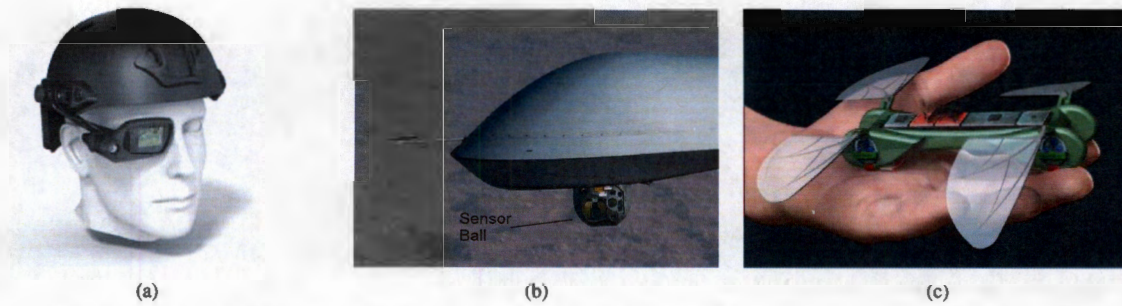


Fig. 2. (a) A smart glass by Vuzix source: [3]; (b) Predator drone source: [5]; (c) Dragonfly-Micro-UAV source: [6].

easily be tricked by false input from a powerful flashlight of an enemy unit. In fact, to the best of our knowledge, currently military CPS and IoT security is limited to protecting the CPS and IoT components networked via traditional means (e.g., RF) or services on the host devices. In other words, securing a networked military CPS and IoT asset means utilizing the same tools and security mechanisms developed for the RF world. However, sensory components in CPS and IoT devices form *sensory channels* that serve as external interfaces to their host systems. Since a significant number of critical functionalities (Figure 2) in the CPS and IoT realms are realized interacting with the real world through these sensory channels, securing the sensory channels is as vital as securing other components of military CPS and IoT assets. Hence, in this paper, we focus on the sensory channel threats to military CPS and IoT assets.

II. SENSORY CHANNEL THREATS

In this section, we describe specific ways of exploiting sensory channels to perpetrate malicious activities against the military CPS and IoT assets.

We primarily envision four different ways to perpetrate malicious activities on CPS/IoT sensory channels. Using these channels, an adversary can (1) trigger existing malware, (2) transfer malware, (3) combine multiple channels to increase the impact of a threat, or (4) leak sensitive information.

In the first threat, the adversary triggers a malicious program existing in the host CPS or IoT device or application where the sensory channel resides. The malicious program is assumed to be loaded into the system's hardware or software without the knowledge of its owner [9]. The malicious program is activated by a specific value or sensory pattern received over the sensory channels. For instance, a malicious program can be triggered over an accelerometer to capture videos, pictures surreptitiously.

The second threat involves utilizing sensory channels to deliver a certain piece of malware to a CPS or IoT device or application. The device could be already compromised or not. A complete malicious code segment or Trojan can be transmitted by the attacker through the sensory channels. As the traditional communication channel (e.g., RF) remains unaffected by this threat, it becomes more difficult to detect or prevent this threat. New Trojans can also be transferred or updated remotely to the compromised CPS or IoT device without being detected.

In the third threat, an adversary can effectively combine more than one sensory channel. Today most of the CPS or IoT devices are manufactured with more than one sensor. For instance, a military armored suit utilize a number of different environment-monitoring sensors (e.g., optical, acoustic, seismic, temperature). Hence, a plausible and a more complicated possible scenario we envision in the third threat is the combination of more than one sensory channel to increase the impact of one channel. In this case, an adversary can combine the sensory channels to increase the effective rate that can be achieved while delivering malware. Furthermore, an adversary can bundle the traditional communication channel with the sensory channels to increase the impact of the damage.

Finally, *in the fourth threat*, an adversary may passively observe the sensitive information leaked through the sensory channels with or without intention.

III. CONCLUSION

In this short position paper, we focused on threats to military CPS and IoT assets through their sensory channels. The sensors on host CPS and IoT devices and applications effectively form the sensory channels. We specifically articulated how a malicious entity can target military CPS and IoT with four different methods exploiting the sensory channels.

REFERENCES

- [1] Samantha Murphy Kelly, "Experts: Internet of things and wearables will dominate by 2025, <http://mashable.com/2014/05/14/pew-iot-study>," May 2014.
- [2] Wikipedia, "Netops, <http://en.wikipedia.org/wiki/NetOps>," February 2015.
- [3] Vuzix, "Vuzix smart glass, http://www.vuzix.com/augmented-reality/products_star1200xld," February 2015.
- [4] D. Hambling, "<http://www.wired.com/dangerroom/2009/04/army-tests-new/#more>," *Wired*, 2009.
- [5] Kashmir Hill, "Drone industry promises not to be evil, <http://www.forbes.com/sites/kashmirhill/2012/07/06/drone-industry-promises-not-to-be-evil/>," July 2012.
- [6] Techject Inc. The Robot Dragonfly, "The robot dragonfly, <http://techject.com/>," February 2015.
- [7] A. S. Uluagac, V. Subramanian, and R. Beyah, "Sensory channel threats to cyber physical systems: A wake-up call," in *2014 IEEE Conference on Communications and Network Security (CNS) (IEEE CNS 2014)*, San Francisco, USA, Oct. 2014.
- [8] V. Subramanian, S. Uluagac, H. Cam, and R. Beyah, "Examining the characteristics and implications of sensor side channels," in *Proc. of IEEE ICC*, June 2013.
- [9] Office of The Secretary of The Department of Defense, Defense Science Board Task Force on Resilient Military Systems, "Resilient military systems and the advanced cyber threat final report," Jan. 2013.

April 1, 2015

NATO STO U.S. National Coordinator
OASD (R&E)/International Technology Programs
4800 Mark Center Drive, Suite 17D08
Alexandria, VA 22350-3600

Dear Sir/Madam:

I am submitting a short position paper entitled "Sensory Channel Threats to Military CPS and IoT Assets" authored by A. Selcuk Uluagac of Florida International University, Miami, Florida, USA. Please accept this as a candidate for the program in the IST-128 Workshop on Cyber Attack Detection, Forensics and Attribution for Assessment of Mission Impact.

Please note the following:

- The work described in this short paper is cleared for presentation to NATO audiences (i.e., approved for public release).
- The paper is technically correct.
- The work is not supported by any government agency.
- The paper is NATO/PfP Unclassified; and
- The paper does not violate any proprietary rights.

I believe that this short paper has a great potential to contribute to the IST-128 Workshop on Cyber Attack Detection, Forensics and Attribution for Assessment of Mission Impact.

Thank you very much for considering this paper and I look forward to receiving your feedback.

Sincerely,

A. Selcuk Uluagac, PhD

Cyber-Physical Systems Security Lab (CSL)
Electrical and Computer Engineering Department
Florida International University
10555 West Flagler Street, EC 3141
Miami, FL 33174, USA

Email: suluagac@fiu.edu
Phone: 1-305-348-3710
Fax: 1-305-348-3707

Group Web: <http://csl.fiu.edu>

Personal Web: <http://web.eng.fiu.edu/selcuk>

Mission Assurance as a Function of Scale

Pierre Trepagnier and Alexia Schulz

Cyber Systems and Operations
MIT Lincoln Laboratory
Lexington, MA 02420

Abstract—Since all Department of Defense (DoD) missions depend on cyber assets and capabilities, a dynamic and accurate cyber dependency analysis is a critical component of mission assurance. Mission analysis aims to identify hosts and applications that are “mission critical” so they can be monitored, and resources preferentially allocated to mitigate risks. For missions limited in duration and scale (tactical missions), dependency analysis is possible to conceptualize in principle, although currently difficult to realize in practice. However, for missions of long duration and large scale (strategic missions), the situation is murkier. In particular, cyber researchers struggle to find technologies that will scale up to large numbers of hosts and applications, since a typical strategic DoD mission might expect to leverage a large enterprise network. In this position paper, we argue that the difficulty is fundamental: as the mission timescale becomes longer and longer, and the number of hosts associated with the mission becomes larger and larger, the mission encompasses the entire network, and mission defense becomes indistinguishable from classic network defense. Concepts generally associated with mission assurance, such as fight-through, are not well suited to these long timescales and large networks. This train of thought leads us to reconsider the concept of “scalability” as it applies to mission assurance, and suggest that a hierarchical abstraction approach be applied. Large-scale, long duration mission assurance may be treated as the interaction of many small-scale, short duration tactical missions.

I. INTRODUCTION

The Department of Defense (DoD) recognizes that all defense missions today depend on cyber infrastructure. The 2010 Quadrennial Defense Review finds that [7] “A failure by the Department to secure its systems in cyberspace would pose a fundamental risk to our ability to accomplish defense missions today and in the future.” The role of cyber dependencies in providing mission assurance has inspired multiple studies and technology development efforts [2], [4], [5], [6], [8], [9]. The DoD must be able to guarantee that it can continue accomplishing critical missions, even in the face of degraded or disabled cyber infrastructure. Identifying the Cyber Key Terrain (C-KT), i.e. those cyber assets necessarily used in mission execution, is a vital ingredient needed to provide such guaranteed mission assurance.¹

There is a divide in the literature regarding the best strategy for identifying the C-KT of a particular mission, and mapping out its network dependencies. Methodologies tend to fall in one of two basic classes: process driven mapping and artifact driven mapping. Process driven mapping makes heavy use of subject

matter experts, and is typically manual and time consuming. Artifact driven mapping leverages usage data and lends itself more readily to automation, but the data frequently lacks sufficient context to reliably identify the C-KT. The proponents of both methodologies are concerned with the ability to scale up to enterprise-scale networks. Of particular concern is that dependency maps of large numbers of hosts, over very long timescales, tend to be difficult to convey succinctly. They often produce a deluge of data which suffers from the “hairball” problem when visually represented.

DoD missions can exist at the strategic, operational or tactical levels. In general, strategic and operational missions are conducted over longer timescales, and are much broader in scope than tactical missions. In this position paper we explore the hypothesis that strategic and operational missions are dependent on, and to a great degree comprised of, sub-missions conducted at the tactical level. Thus effective mission assurance at every layer of the hierarchy depends on the ability to map tactical level missions with fidelity. This is particularly pertinent to the cyber domain. It may be misguided to focus entirely on techniques and visualizations that scale up to enterprise network scale, or are capable of processing data volumes from extended periods of time. Indeed such techniques may over-aggregate and not provide sufficient situational awareness to identify and mitigate risks to the mission.

Although this discussion takes place in the context of the DoD, all of the conclusions can be generalized to apply to the civilian arena. All large organizations include missions that can be described as tactical, operational and strategic.

II. TIMESCALE

Strategic, operational and tactical missions are conducted over distinct characteristic times. Strategic missions capture the essential role of the organization; e.g. the mission of DoD is to provide the military forces needed to deter war and to protect the security of our country [1]. Because of this, strategic missions are executed continuously rather than on a short timescale, and the mission definition evolves very slowly if at all. In contrast, tactical missions tend to comprise a specific set of military actions with a well defined goal that is easily measured; e.g. conduct an airstrike against a particular target. The duration of tactical missions is generally short, although the mission can be repeated multiple times. Tactical missions are defined and executed based on specific military actions that need to be taken, so the mission definitions are variable and are often not known in advance. Finally, operational missions involve resource allocation and the integration of tactical missions to achieve strategic ends [3]. The timescales for operational missions are generally long, but the mission definition may evolve more swiftly than that of a strategic mission.

¹The DoD definition of key terrain in general is “Any locality, or area, the seizure or retention of which affords a marked advantage to either combatant.” The cyber version of this would also include assets that enable the adversary to execute its mission against the U.S. For the purposes of this paper, however, we have adopted a more restrictive definition focused on mission assurance.

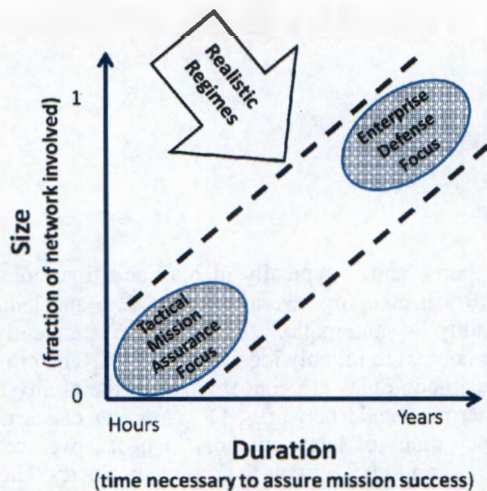


Fig. 1. Figure 1. Tactical mission assurance involves relatively few hosts and short timescales; enterprise defense involves many hosts and long to indefinite timescales.

According to USAF doctrine, while the resulting effects may be described as operational or strategic, military actions occur almost entirely at the tactical level [3]. This is particularly true in cyberspace. While cyber assets are frequently used to provide information and command and control in support of strategic and operational missions, the delivery of information, key applications, services, and command channels occurs entirely at the tactical level. This fact generates an argument for shifting the focus of dependency mapping efforts to providing mission assurance at the tactical level.

Another reason to make this shift is that certain central elements of mission assurance are easier to define and measure at the tactical level than at the operational or strategic levels. The ability to “fight through” a contested cyberspace is a concept that only applies for missions of finite duration; one cannot fight through to infinity.

III. NUMBER OF HOSTS

Strategic and operational missions use a larger fraction of the total hosts on the network than tactical missions. Indeed, an enterprise network exists to serve the strategic missions of the organization. In contrast, tactical missions are generally supported by a small fraction of total network. Good network hygiene dictates that if a host is not supporting any organizational mission, it needlessly presents extra attack surface to adversaries and should be removed. But network hygiene is distinct from mission dependency mapping; the central aim of mission dependency mapping is to identify a restricted set of hosts (as a fraction of total network) critical to a particular effort. If the number of hosts necessary to prosecute a mission approaches the size of the network, mission defense is indistinguishable from classic network defense. In practice, number of hosts and timescale (discussed above) are correlated, depicted schematically in Figure 1.

Another central element of effective mission assurance at the operational and strategic levels incorporates well defined

Courses of Action (COAs) designed to help decision makers react to evolving priorities and risks. Mapping the COA dependencies independently is critical. In a contested cyber environment one cannot defend every asset. Limited resources need to be allocated to defend highest priority cyber terrain, based on tactical decisions regarding which COA is being pursued in support of the operational or strategic mission.

IV. DISCUSSION

The import of the arguments presented here is that mission assurance software need not “scale” to the size of a global enterprise, as the term scaling is usually defined. Visualizations and algorithms need not work for thousands of hosts. If thousands of hosts are present in the dependency map of an operational or strategic mission, with little or no fidelity in the mapping of the tactical importance of these hosts, it will be difficult for mission defenders to know which hosts should be monitored. Such a dependency map will not help them correctly prioritize the allocation of resources, rather it will be an illegible hairball, and be ignored.

Enterprise scale mission assurance is instead achieved by hierarchical decomposition into tactical missions, each associated with a particular COA. It is important to explore the validity of modeling strategic or operational missions as entirely composed of missions at the tactical level, with the overarching mission being decomposed into sub-missions, and sub-missions decomposed into sub-sub-missions, and so forth. At each mission level, as much detail as possible of the level below is abstracted away, leaving only those details which are necessary to maintain fidelity of mission interactions. In this manner, the problems associated with scaling and data deluge are minimized. However, effective models of mission assurance for operational and strategic missions will necessarily involve retaining enough fidelity to capture the complex interactions possible between multiple tactical building blocks [10]. Determining the minimum necessary level of fidelity is an important area for future investigation.

V. CONCLUSIONS AND RECOMMENDATIONS

In summary, we are asserting two major propositions. First, in the cyber domain, crucial mission assurance constructs such as cyber key terrain and fight-through are meaningful for tactical missions involving limited time and a small fraction total network resources, but cease to be useful for enduring missions which utilize a large fraction of cyber resources. In the latter case, mission assurance simply degenerates into classic network defense and network hygiene. Second, that large, enduring, strategic missions may in fact be decomposed hierarchically into many small tactical ones, and that by so doing problems of scaling, data deluge, and visualization (the hairball problem) are minimized by dropping complexity between layers of the hierarchy. The outstanding problem becomes determining the minimum fidelity necessary in the dependency mapping of tactical missions and sub-missions to maintain accurate models of complex system interactions between the tactical building blocks. Our recommendations are to focus near term efforts on developing technology for the swift and accurate mapping of tactical missions, with a longer term focus on modeling their complex interactions to assure larger scale missions.

ACKNOWLEDGEMENTS

The authors wish to thank Chris Degni, Jeffrey Gottschalk, and Reed Porada for helpful comments on an early draft.

This work is sponsored by the by Department of the Air Force under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

REFERENCES

- [1] U. S. Department of Defense, <http://www.defense.gov/about>.
- [2] *Camus: Automatically Mapping Cyber Assets to Missions and Users*, MILCOM 2009. IEEE, Oct 2009.
- [3] *Basic Doctrine Volume 1*. Curtis Lemay Center for Doctrine Development and Education, 2011.
- [4] *A systems engineering approach for crown jewels estimation and mission assurance decision making*. IEEE, April 2011.
- [5] *NSDMiner: Automated Discovery of Network Service Dependencies*, IEEE International Conference on Computer Communications, Orlando, FL, March 2012. IEEE.
- [6] R. Elder. Defending and operating in a contested cyber domain,". *Air Force Scientific Advisory Board, Winter Plenary*, 2008.
- [7] R. Gates. *Quadrennial Defense Review Report (Feb. 2010)*. DIANE Publishing Company, 2010.
- [8] E. Peterson. Dagger: Modeling and visualization for mission impact situation awareness. Manuscript submitted to 2015 IEEE International Symposium on Technologies for Homeland Security, 2015.
- [9] J. Watters, S. Morrissey, D. Bodeau, and S. C. Powers. The risk-to-mission assessment process (riskmap): A sensitivity analysis and an extension to treat confidentiality issues. *The Institute for Information Infrastructure*, July 2009.
- [10] W. Young and N. G. Leveson. An integrated approach to safety and security based on systems theory. *Communications of the ACM*, 57(2):31-35, 2014.

NATO STO U.S. National Coordinator
OASD (R&E)/International Technology Programs
4800 Mark Center Drive, Suite 17D08
Alexandria, VA 22350-3600

April 6, 2015

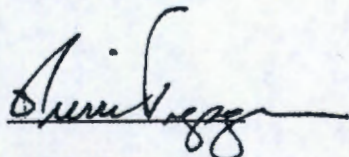
RE: *Mission Assurance as a Function of Scale* (Position Paper)

Dear Mr. Uribe:

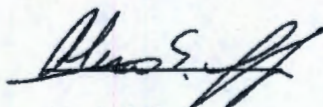
We, the authors of the enclosed paper do state:

- The work described in this paper is cleared for presentation to NATO audiences (i.e., Approved for Public Release, Distribution Unlimited)
- The paper is technically correct
- The work was sponsored by the Department of the Air Force. We attest that the organization (AFLCMC/XZCC) is aware of submission and has approved it for public release
- The paper is NATO/PfP Unclassified
- The paper does not violate any proprietary rights.

Very truly yours,



Pierre Trepagnier
Technical Staff



Alexia Schulz
Technical Staff

MIT Lincoln Laboratory
244 Wood Street
Lexington, MA 02420

ATTACHMENT: MIT LL Release Review Form, Request 105430

Cyber-Attack as a Contest Game: Modeling the Economic Impact

Alexander Alexeev

Odessa State Ecological University / Indiana University
UKRAINE / USA

aalexeev@indiana.edu

Kerry Krutilla

Indiana University, Bloomington
USA

krutilla@indiana.edu

ABSTRACT

We apply a contest-game theoretic framework for modelling the economic impact of a cyber-attack. In the model the attacker/defender allocate available resources and efforts to maximize gain/minimize loss from the attack. Among its useful features, the parsimonious model allows for the assessment of the asymmetry in the effectiveness of the resource use, different scale for gains and losses, and the non-zero probability of the unknown vulnerability to be exploited in the attack. A Nash solution in pure strategies is demonstrated and analysed.

1.0 INTRODUCTION

Governmental security systems face cyber-threats from many sources, including solo hackers, criminal organizations, and intelligence services of other nations. In this research, we focus on the government-to-government rivalries. A model is developed that shows the economically efficient level of resources a defending nation should commit to its cyber security system given the expected economic loss associated with a successful attack, and the counter actions of a rival. This research is effectively a form of benefit cost analysis conducted within the context of a strategic rivalry between countries.

Although game theory methods have been used in the general field of terrorism study for the past two decades (See Sandler and Sequeira, 2009), the application to cyber security policymaking has been relatively recent (See Roy et al 2010 and Lazka et al 2014 for reviews). Our contribution is to show the way economically efficient defensive investments should reflect the gains and losses of a successful attack, and the parameter values of the functional form relating resource commitments in defence and attack to outcome probabilities.

We begin in the next section with a discussion of the modelling motivation and formulation. The following section solves the model, and the next two sections detail the economically rational level of resource commitments of both the attacking and the defending nation. The following section discusses the possibility of decision-making noise that affects the equilibrium solutions for the model. The final section offers some

concluding remarks and suggestions for additional research.

2.0 THE MODEL

We assume a rivalry between the intelligence services of two countries in which a defender country faces a cyber-attack from the rival. The rivals are relatively symmetrically positioned in terms of level of technology and available resources, and the probability of a successful attack is reasonably high. This context contrasts with the asymmetric threats with low probability outcomes, such as those posed by terrorists organizations attempting to launch an attack on the territory of a well defended country.

In the benchmark model developed in this paper, it is assumed that both intelligence services are rational -- as rationality is defined in conventional economic models -- and that the rivals are informed about each other's actions. The rationality assumption is reconsidered below. In this version of the paper, we also make the simplifying assumption that the game is not repeated.

Specifically, we assume a one-stage, simultaneous move game in which the one country ("the attacker") attacks the cyber-infrastructure of the other country ("the defender"). The probability that the attack succeeds reflects the resources committed by both the attacker and the defender. The goal of the attacker is to maximize their expected gains from attack, while the objective of a defender is to minimize their expected losses. The solution concept is Nash equilibrium. Following Krutilla and Alexeev (2012) the model is represented as follows:

$$\max_{R_A} P_A = G_A p(R_A, R_D) - R_A, \quad (1)$$

$$\min_{R_D} P_D = L_D p(R_A, R_D) + R_D, \quad (2)$$

The variable R_A and R_D are the resource commitment by the attacker and the defender respectively, and P_A and P_D are the expected net-pay offs of the two revivals. The monetized value of the attacker's gain G_A and defender's loss L_D are both exogenous variables. G_A can be thought of as the monetized utility value that the attacker derives from the damage they create; L_D is the monetized utility value of the damage to the defender. Without loss of significant generality, we express the relationship between G_A and L_D as $G_A = gL_D$, with $0 \leq g$. The g parameter can be seen as an attacker's unit valuation of a dollar of damage they create.

The term $p(R_A, R_D)$ in equations (1)-(2) denotes an attack success function that gives the probability of the attack's success as a function of the attacker's and defender's resource commitments. The functional form used for $p(R_A, R_D)$ is based on modified contest success function commonly used to model rent-seeking contests in the political economy literature (See Tullock 1980 and Glachant 2005):

$$p(R_A, R_D) = p_0 + \frac{R_A^r}{R_A^r + aR_D^r} \quad (3)$$

The term p_0 , $0 < p_0 \leq 1$, represents the exogenous probability of a successful attack. A non-zero initial probability may exist due to exogenous technical change in the form of new information about a

system's vulnerability that an attacker can exploit with virtually no investment. The necessary domain restriction is: $p_0 \in p(R_A, R_D) \in [1, \infty)$.¹ Turning to the other parameters, $a \in (0, \infty)$ represents asymmetries in the relative effectiveness of rivals' resource commitments. The range $a < 1$ imply relatively greater effectiveness of the defender's resource commitments, while $a > 1$ implies relatively greater effectiveness for the attacker. The r parameter,² with $r \in (0, \infty)$, represents the returns to attacking and defending resource commitments. Increasing its value gives relatively more weight in the outcome probability to whichever rival devotes more resources to the contest.

3.0 RESULTS

Substituting (3) into (1) and (2) and solving for (R_A^*, R_D^*) gives candidates for Nash equilibria in pure strategies. The solutions turn out to be:

$$r_A^* = \frac{a r g^{r+1}}{(g^r + a)^2} \tag{4}$$

$$r_D^* = \frac{a r g^r}{(g^r + a)^2} \tag{5}$$

The new left-hand side variables are $r_A^* \circ R_A^* / L_D$; $r_D^* \circ R_D^* / L_D$. That is, the left-hand side variables give the ratio of each of the rivals resource commitments to the defenders damage loss. The right-hand side gives all of the exogenous parameters in the model. Figure 1 below illustrates the optimal resource commitments as a function of the models parameters.

Note that if (5) is divided by (4), the result is the simple expression:

$$\frac{r_D^*}{r_A^*} = \frac{R_D^*}{R_A^*} = \frac{1}{g} \tag{6}$$

This implies that the ratio of efforts and resources devoted by the defender to that by attacker is inverse proportional to g -- again, the unit valuation by the attacker of a dollar of damage to the defender -- whatever the absolute gains, G_A or loss L_D associated with a successful attack. Given the assumption that $0 \leq g$, the defender's resource commitments (R_D^*) will always be greater in equilibrium than the attacker's (R_A^*) when the attacker underestimates its gain ($0 \leq g < 1$), and vice versa when the damage value to the attacker is overestimated ($1 \leq g$).

¹ Formally, (3) must be written as $p(R_A, R_D) = \min\left\{\frac{G_A}{L_D}, p_0 + \frac{a}{g} + a \left(\frac{R_D}{R_A}\right)^r\right\}$.

² See Baye et al 1994, Nitzan 1994, Perez-Castrillo Verdier 1992.

Cyber-Attack as a Contest Game: Modeling the Economic Impact

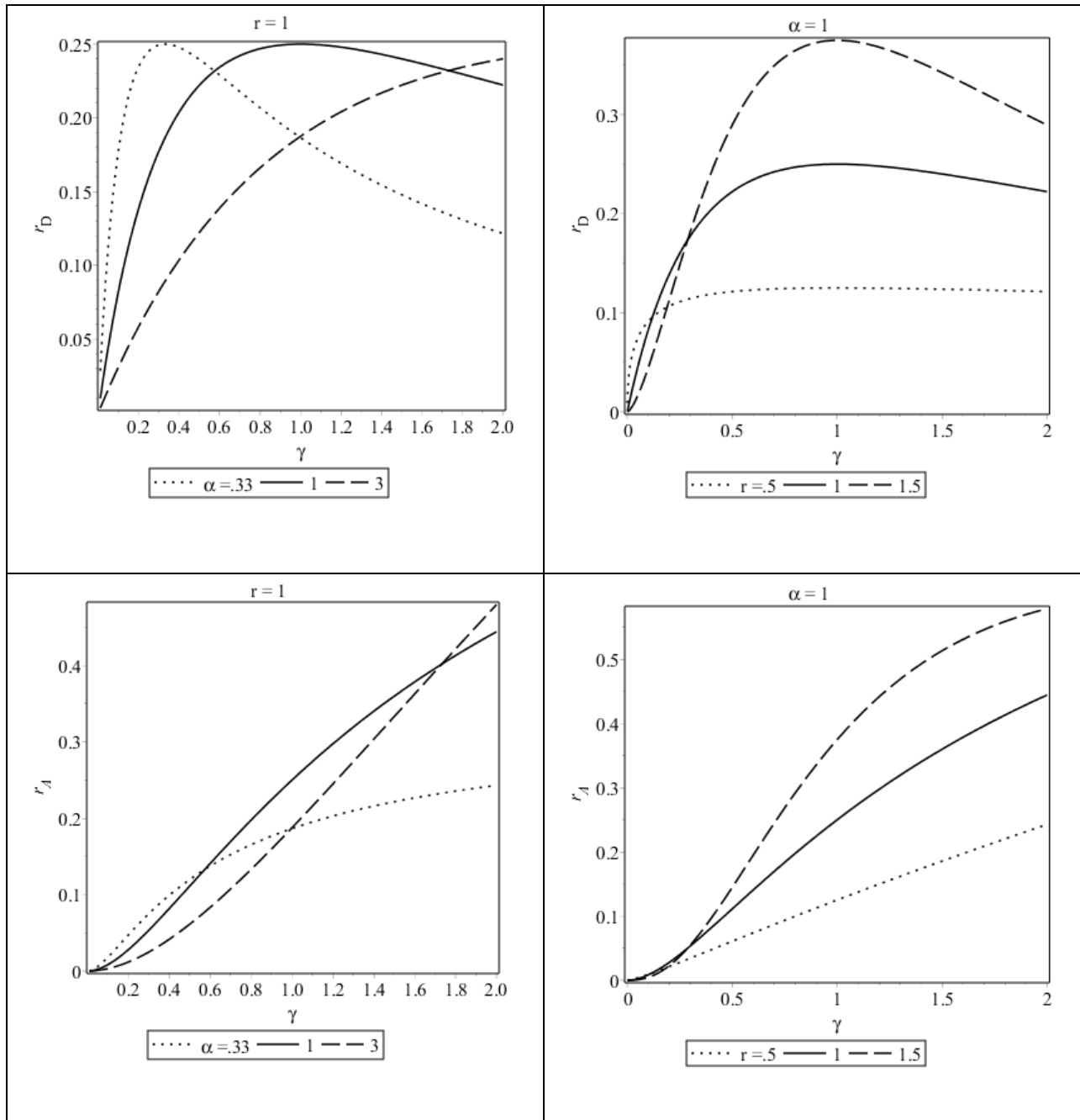


Figure 1. Optimal Resource Commitments, as a Function of the Model's Parameters for the Defender (upper cells) and for the Attacker (bottom cells)

Substituting (6) into (3) gives the reduced form probability for the success of the attack:

$$p(R_A^*, R_D^*) = p_0 + \frac{g^r}{g^r + a} \tag{7}$$

The probability of the attack’s success, $p(R_A^*, R_D^*)$ increases in g and declines in a -- again, the latter is relative effectiveness of resources use by the defender. The impact of r on $p(R_A^*, R_D^*)$ gives rise to increase of the probability with r given a fixed. The higher return to attack – the higher probability of attack success. Figure 2 below shows the probability of successful attack as a function of the models parameters.

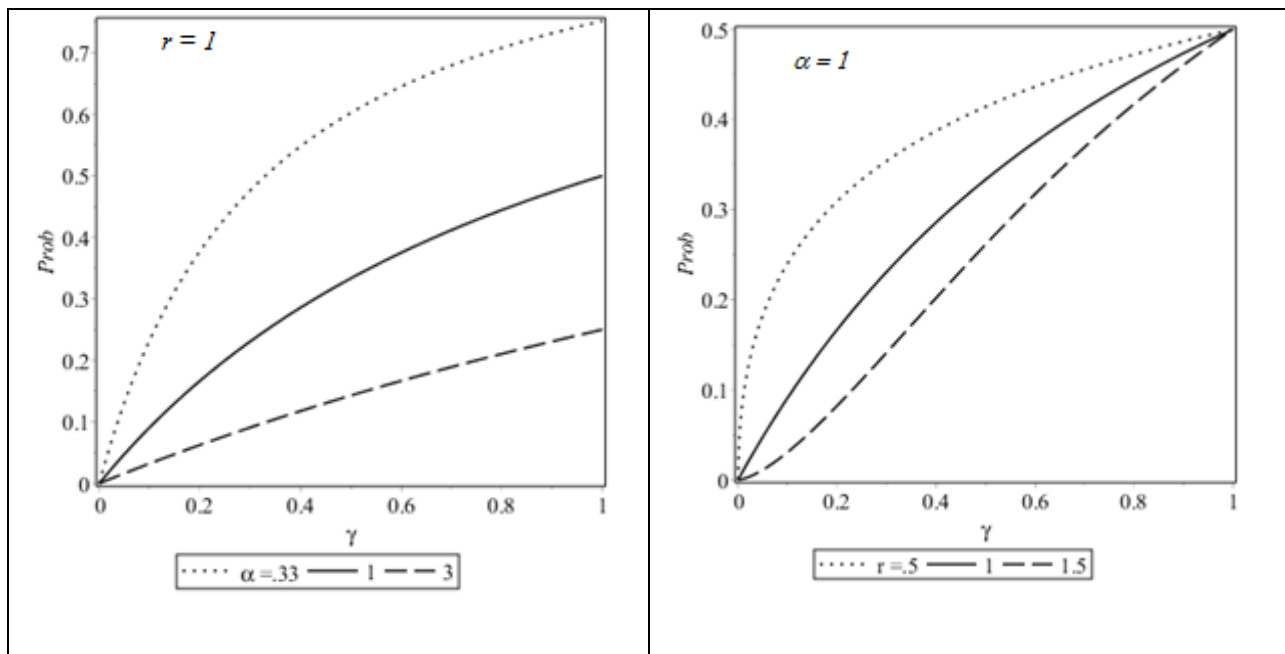


Figure 2. Probability of Successful Attack Curves, as a Function of the Model’s Parameters

4.0 RATIONALITY ASSUMPTION AND DECISION MAKING NOISE

Although presumably government intelligence services should be acting analytically, there are a number of rationales behind relaxing ubiquitous economic assumption that they would chooses the best strategy in an optimizing game theoretic framework. Both laboratory experiments and empirical observation often reveal deviation from the strictly rational behaviour, including around cyber-security issue (see, e.g. Bada et al (2015), Yang et al (2015) among others). There are several modelling frameworks for the bounded rationality. All the approaches assume that the agent choses not the best strategy but a “reasonably good” strategy that deviates from optimality with the probability declining with the deviation magnitude. For example, Wall (1993) develops and implements dynamic and adaptive models that combine satisficing behaviour with learning and adaptation through environmental feedback. This a sequential decision making with one alternative strategy at time, with search strategies based on learning and adaptation. Quantal Response Equilibrium (QRE) is another approach (see, e.g. Sheremeta (2015), An et al (2013)). Another framework has

Cyber-Attack as a Contest Game: Modeling the Economic Impact

been developed by Amigashi (2006). His contest success function originally discovered by Dasgupta and Nti (1998) adds a “noise parameter” in the decision making process. In cyber-security applications, it might measure false-positive alarms in the intrusion detection system. In this context, the contest success function (shown here for the attacker) has the following form:

$$p(R_A, R_D) = \frac{R_A^r + l}{R_A^r + aR_D^r + 2l} \tag{8}$$

where l is the noise parameter, and a, R_A, R_D are defined in (3). When, $\lambda=0$, the decisions are defined by the Nash equilibria) as calculated in y (4)-(5). As l increases, the solutions departs farther from the Nash equilibria, and becomes less sensitive to the value of the resources invested into attack/defense. As $l \rightarrow \infty$ the choice of strategy is absolutely random, that is $p(R_A, R_D) = 0.5$ regardless of what actions the defending country and attacking country take. The solutions of the contest game (1), (2) and (8) include parameter l and can be expressed as following :

$$r_A^* = \frac{g(g - 2l)a - a^2l - g^2l}{(a + g)^2}$$

$$r_D^* = \frac{(g - l)a^2 - 2agl - g^2l}{a(a + g)^2} \tag{9}$$

Figure 3 below illustrates the optimal resource commitments as a function of the models parameters and noise parameter l .

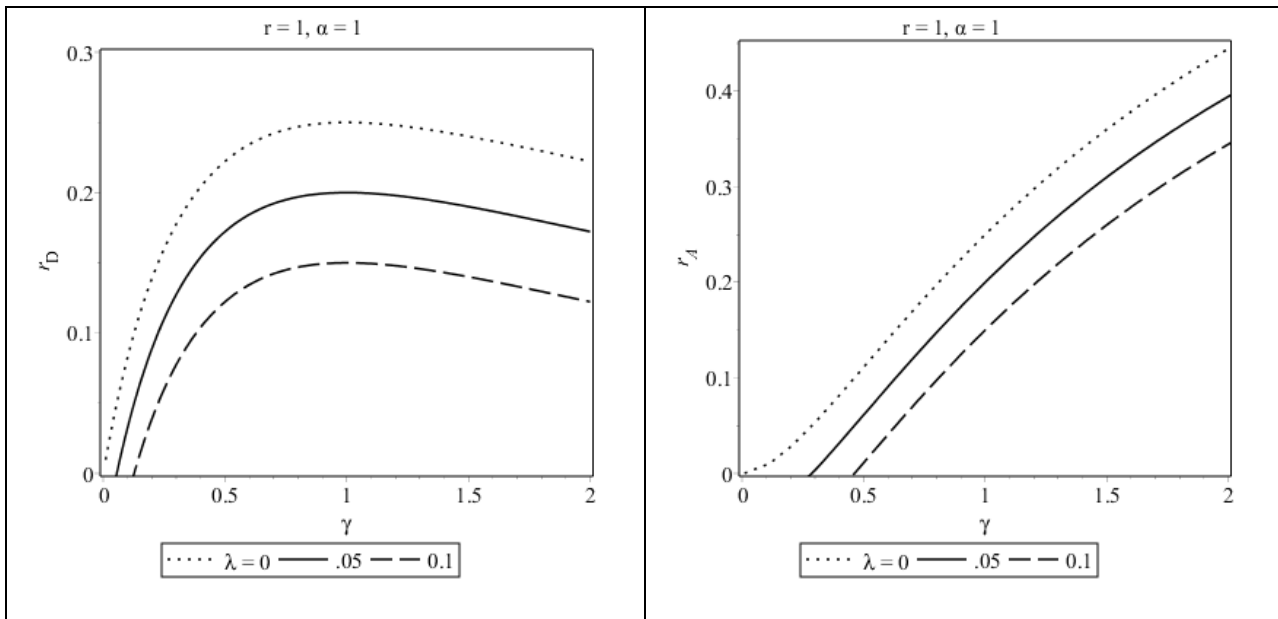


Figure 3. Optimal resource commitments, as a Function of the Noise Parameter for the defender (left) and for the attacker (right)

Note that ratio r_D^* / r_A^* defined in (6) takes the following form:

$$\frac{r_D^*}{r_A^*} = \frac{R_D^*}{R_A^*} = \frac{(g - l)a^2 - 2agl - g^2l}{a(g(g - 2l)a - a^2l - g^2l)} \quad (10)$$

At $\lambda=0$ the decisions (9) are defined by the Nash equilibria as calculated in (4)-(5), and ratio $r_D^* / r_A^* \approx g^{-1}$ defined in (6). At the other extreme, at $l \approx \infty$, the ratio(10) of the resource costs is $r_D^* / r_A^* \approx a^{-1}$. Recall, parameter a represents asymmetries in the relative effectiveness of rivals' resource commitments, and implies relatively greater effectiveness of the defender's resource commitments while $a < 1$, and vice versa while $a > 1$. In perfectly noisy environment $r_D^* / r_A^* \approx a^{-1}$ implies that the ratio of efforts and resources devoted by the defender to that by attacker is inverse proportional to a and independent from g . In other words, the ration does not depend on either absolute values of gains, G_A , or loss, L_D , associated with a successful attack nor on their ratio. Note, in (9) and (10), noise parameter, l , is dimensionless and scaled in units of L_D ; $r = l$ assumed for simplicity. It can routinely be shown that both $\frac{\partial r_D^*}{\partial l} < 0$ and $\frac{\partial r_A^*}{\partial l} < 0$. This coincides with the conventional wisdom. As a strategy departs from optimality due to bounded rationality/increasing noise in the system, both the probability of the attack's success and payoffs become less and less sensitive to the resource cost allocated for the attack/defense, and, consequently, less resources is required for the optimal strategy.

5.0 CONCLUSION AND FUTURE RESEARCH

We have used a contest-game theoretic framework to the model a strategic contest between the intelligence services of two countries, where the one country attempts to penetrate the cyber system of the other, and the country on the receiving end of the cyber-attack attempts to defeat the attack. The attacker and defender allocate resources to maximize their gains and minimize their losses, respectively, taking into account the actions of each other. The model used to describe this interaction represents the effects of asymmetry in the effectiveness of the resource commitments, different scales for gains and losses, and non-zero probability of the unknown vulnerability to be exploited in the attack. A Nash solution in pure strategies is used to evaluate a dependence of the initial probability of the successful attack on the measure of the attack's detrimental effect.

Conceptually, the behaviour of intelligence services should be assessed in a generalized contest model that incorporates a measure risk-averseness of the attacker/defender, bounded rationality, decomposition of the risk attitude into systematic and idiosyncratic components, and judgmental bias -- among others. Additionally, the actors in the contest will be interacting over multiple periods. We are beginning to incorporate these features into the model in our ongoing research program.

References

- [1] Amegashie, J. A. (2006). A contest success function with a tractable noise parameter. *Public Choice*, 126(1-2), 135-144.
- [2] An, B., Ordóñez, F., Tambe, M., Shieh, E., Yang, R., Baldwin, C., ... & Meyer, G. (2013). A Deployed Quantal Response-Based Patrol Planning System for the US Coast Guard. *Interfaces*, 43(5), 400-420.
- [3] Baye, M.R., D. Kovenock, and C.G. de Vries. (1994). The solution to the Tullock rent-seeking game when $R > 2$: mixed-strategy equilibria and mean dissipation rates. *Public Choice* 81(3): 363-380.
- [4] Bada, M., Sasse, A., & Nurse, J. R. (2015). Cyber Security Awareness Campaigns: Why do they fail to change behaviour? *Procs. Int. Conf. on Cyber Security for Sustainable Society*, 118–131
- [5] Dasgupta, A., & Nti, K. O. (1998). Designing an optimal contest. *European Journal of Political Economy*, 14, 587–603.
- [6] Glachant, M. (2005). *Voluntary agreements in a rent-seeking environment*. The Handbook of Environmental Voluntary Agreements 43(2): 49–63.
- [7] Krutilla, K., & Alexeev, A. (2012). The Normative Implications of Political Decision-Making for Benefit-Cost Analysis. *Journal of Benefit-Cost Analysis*, 3(02), 1-36.
- [8] Laszka, A., Felegyhazi, M., & Buttyan, L. (2014). A survey of interdependent information security games. *ACM Computing Surveys (CSUR)*, 47(2), 23.
- [9] Nitzan, S. 1994. Modeling rent-seeking contests. *European Journal of Political Economy*. 10: 41-60.
- [10] Perez-Castrillo, D., and T. Verdier. 1992. A general analysis of rent-seeking games. *Public Choice* 73: 335-350
- [11] Roy, S., Ellis, C., Shiva, S., Dasgupta, D., Shandilya, V., & Wu, Q. (2010). A survey of game theory as applied to network security. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on* (pp. 1-10). IEEE.
- [12] Sandler, T., & Siqueira, K. (2009). Games and Terrorism Recent Developments. *Simulation & Gaming*, 40(2), 164-192.
- [13] Sheremeta, R. M. (2015). 10. Behavioral dimensions of contests. *Companion to the Political Economy of Rent Seeking*, 150.
- [14] Tullock, G.(1980). *Efficient rent seeking*. In *Toward a theory of the rent seeking society*, edited by James M. Buchanan, Roubert D. Tollison, and Gordon Tullock. College Station, Texas. Texas A&M University Press, 131-76.
- [15] Wall, K. D. (1993). A model of decision making under bounded rationality. *Journal of Economic Behavior & Organization*, 20(3), 331-352
- [16] Yang, L., Toubia, O., & De Jong, M. G. (2015). A Bounded Rationality Model of Information Search and Choice in Preference Measurement. *Journal of Marketing Research*, 52(2), 166-183.



UNCLASSIFIED

Cyber-Attack as a Contest Game: Modeling the Economic Impact



Position Paper

Cyber Risk Analysis of CIS-Dependent Missions:
A Modeler's Perspective on Preparing for,
Detecting, and Responding to Cyber Attacks

IST-128 Workshop on Cyber Attack Detection, Forensics and Attribution
for Assessment of Mission Impact

Matthew H. Henry*, David R. Zaret, J. Ryan Carr, J Daniel Gordon
Johns Hopkins University Applied Physics Laboratory
11100 Johns Hopkins Road, Laurel, MD 20723-6099

April 1, 2015

Modern warfare is increasingly dependent upon resources, connections, and interactions in the domain that Gibson famously dubbed *cyberspace*.¹ It follows, then, that the deepening reliance on electronic data, networks, and computing resources presents a valuable target to adversaries. Therefore, combatant commanders should seek to protect their Communications and Information Systems (CIS) from corruption and denial by saboteurs in much the same way that protecting lines of supply and communication was at the forefront of every general's mind in the massive land campaigns of the last century.²

Conflict is the result of competing, imperfectly informed decision makers applying resources against targets. This is equally true of adversarial interactions in cyberspace, where decision makers can be human or artificial agents; targets include data and platforms used by CIS; resources include exploits, credentials, and data manipulation scripts; and partial observability is the result of imperfect and limited aperture sensors. To better prepare for, detect, and respond to attacks in cyberspace, we must seek to understand not only what an adversary might do in this space but also how an

*corresponding author: matthew.henry@jhuapl.edu, (240) 228-2585

¹W. Gibson, *Burning Chrome*, *Omni*. July, 1982, pp. 72-77,102-104.

²Thanks to my colleague Chuck Crossett for articulating this useful analogy.

attack might proceed in the context of different opportunities and obstacles presented to the attacker.

Opportunities and obstacles can be passive and static, in which case they present an adversary with fixed terrain, or they can be responsive and dynamic, in which case they present an adversary with a much less predictable landscape. Greater insight into the alternatives available to an adversary and the associated outcomes of different adversary choices under different circumstances will help the CIS Defender to extrapolate comprehensive attack awareness from sparse detection events so that countermeasures can be more effectively deployed. Moreover, *a priori* understanding of how attacks in the aggregate might proceed in the context of different types of opportunities and obstacles will help CIS system engineers make better-informed decisions about architectural design choices, selection and placement of sensors, implementation of intrusion detection and prevention safeguards, and institution of operational practices and training programs.

The key to achieving these needed insights is good modeling and model-based analysis. Our purpose here is to briefly illustrate the relative benefits and shortcomings of one of the most prevalent and useful modeling paradigms in current practice, graph-based analysis, and then contrast it with a new approach that explicitly accounts for adversary decision processes and the effects of partially observed attack state spaces when modeling conflicts in cyberspace.

Graph-based modeling techniques compute measures of cyber attack state reachability, where the attack state is typically described by the set of resources accessible by the attacker at any stage of the attack.³ The advantages of this approach include a relatively manageable data requirement for constructing the model, repeatably and precisely computable outcomes, and easily interpretable results. Moreover, because these techniques focus largely on graph traversal, where the nodes typically represent system resources and access requirements, finding high-value passive security enhancement opportunities such as firewall rules, access control policies, and so forth, can be straight-forward.⁴

In graph-based analysis, consequence measures for each reachable attack state are used to assess risk. These measures can be estimated using a variety of techniques, including consequence state reachability. Consequence state is described by the set of outcomes that can be induced by the attacker

³cf. K. Ingols et al., Modeling Modern Network Attacks and Countermeasures Using Attack Graphs, *2009 Ann Comp Sec App Conf (ACSAC'09)*, Dec 7-11, 2009, pp.117-126.

⁴cf. M.H. Henry et al., Coupled Petri Nets for Computer Network Risk Analysis, *Intl Journal Critical Infrastructure Protection*, 3(2), 2010, pp. 67-75.

through manipulation of CIS-supported processes via access and control authority afforded by the network resources accessible in the corresponding attack state.⁵ Alternatively, consequence measures can be estimated using decomposition techniques that provide a structure for assimilating and aggregating assessments provided by subject matter experts (SME). This approach decomposes a CIS in terms of data flows, network resources, and mission activities to assert mission consequences due to network resource or data manipulation.⁶

While the aforementioned modeling methods provide valuable insight into the mechanics and potential outcomes of cyber attacks, they are essentially limited to understanding attacks under passive defenses. As such, they provide limited insight into the value of active defensive measures, whether proactive or responsive, for the purpose of informing investments along these lines. We assert that active defenses are critical, particularly in light of the fact that legitimate system users inevitably provide a substantial component of the attack surface by remaining susceptible to social engineering (e.g., spear phishing) and other deception-based intrusion methods. Under these conditions, passive defenses are less effective since activities executed under the auspices of legitimate credentials generally appear to be benign by passive measures.

While researchers and practitioners in the computer network defense community generally agree with this assertion on an intuitive basis, there is little agreement on how best to invest in active defenses. Moreover, there are no credible mature techniques, other than SME intuition, to assess the value of different investments in active defensive measures. Finally, there are no mature tools, other than clever visualization schemes, that provide deep insight into how attacks are playing out when only scarce indicators are available to inform response activities.

We are developing new game theoretic methods that explicitly account for an attacker's decision process in the context of active defenses and partial information so that system architects and engineers can gain insight into the value of these defenses and their associated intrusion detection mechanisms for the purpose of informing broader security investment decisions. Moreover, by explicitly modeling a partially observed state space, we are working toward methods to help defenders infer the true extent of an attack that is

⁵cf. M.H. Henry et al., Evaluating the Risk of Cyber Attacks on SCADA Systems via Petri Net Analysis. *2009 IEEE Conf Tech Homeland Sec HST'09*. May 11-12, 2009, pp.607-614.

⁶cf. T. Llanso & E. Klatt, CyMRisk: An Approach for Computing Mission Risk due to Cyber Attacks, *8th Ann IEEE Sys Conf (SysCon)*. Mar 31-Apr 3, 2014, pp.1-7.

underway when only scarce indicators are available from sensors.

At a high level, our approach is to model the CIS Attacker's decision process as a partially observed stochastic optimization problem in the context of opportunities (e.g., access to a host's resources) and obstacles (e.g., credentials needed to access a host's resources) that the intruder discovers and responds to over the course of the intrusion. At the same time, the CIS Defender may at times detect indicators of an intrusion in the form of host infection and respond by introducing additional obstacles (e.g., isolating and reconstituting an infected host). As such, the interaction constitutes a partially observed stochastic game with finite state space. This work complements other approaches reported in the academic literature.⁷

Two research problems are the focus of our current work. The first is to develop efficient computational methods for identifying near optimal strategies when presented with partially observed state spaces. The necessary estimation of belief measures on the true state quickly overwhelm standard algorithmic approaches based on policy iteration. Techniques borrowed from the artificial intelligence community are proving useful as means to approximate optimal strategies under partial information. The second is to develop reliable parameter estimation techniques from intrusion data, vulnerability databases, and intrusion detection data. A forthcoming Springer volume, expected later this year, will include a more in-depth discussion of our model-based analysis work, its associated research challenges, and the anticipated benefits for improving both strategic risk assessment and tactical situational awareness.

In spite of the challenges, we are confident that approaches such as the one we are pursuing will yield the insights needed to better understand, prepare for, detect, and respond to conflicts in cyberspace. Moreover, we assert that analytic techniques that account for adversary decision processes are necessary to inform strategic and tactical decision-making when the adversary's intentions, maneuvers, and disposition are only partially, and perhaps imperfectly known. This has been the traditional approach to strategy development in other conflict domains, and it applies equally to cyberspace.

⁷cf. S.A. Zonouz et al., RRE: A Game-Theoretic Intrusion Response and Recovery Engine. *2009 Intl Conf Dep Sys & Nets (DSN09)*. Jun 29-Jul 2, 2009, pp.439-448.

APR 3 2015

Cyber Risk Analysis of CIS-Dependent Missions:
A Modeler's Perspective on Preparing for,
Detecting, and Responding to Cyber Attacks

IST-128 Workshop on Cyber Attack Detection, Forensics and Attribution
for Assessment of Mission Impact

Matthew H. Henry*, David R. Zaret, J. Ryan Carr, J Daniel Gordon
Johns Hopkins University Applied Physics Laboratory
11100 Johns Hopkins Road, Laurel, MD 20723-6099

March 31, 2015

Dear Sir or Madam:

The work described in this paper is approved for public release in general and, specifically, is cleared for presentation to NATO audiences. The work was developed under internal research and development funding. The paper is technically correct. The paper is NATO/PfP Unclassified and does not violate any proprietary rights.

Matthew H. Henry

*corresponding author: matthew.henry@jhuapl.edu, (240) 228-2585

Analyzing Mission Impacts of Cyber Actions (AMICA)

Steven Noel, Jackson Ludwig, Prem Jain, Dale Johnson,
Roshan K. Thomas, Jenny McFarland, Ben King
The MITRE Corporation
7515 Colshire Drive, McLean, Virginia, 22102, USA
[snoel, ludwig, pjain, djohnson, rkthomas,
jmcfarland, bking]@mitre.org

Seth Webster, Brady Tello
MIT Lincoln Laboratory
244 Wood Street, Lexington, Massachusetts, 02420, USA
[swebster, brady.tello]@ll.mit.edu

Abstract — This paper describes AMICA (Analyzing Mission Impacts of Cyber Actions), an integrated approach for understanding mission impacts of cyber attacks. AMICA combines process modeling, discrete-event simulation, graph-based dependency modeling, and dynamic visualizations. This is a novel convergence of two lines of research: process modeling/simulation and attack graphs. AMICA captures process flows for mission tasks as well as cyber attacker and defender tactics, techniques, and procedures (TTPs). Vulnerability dependency graphs map network attack paths, and mission-dependency graphs define the hierarchy of high-to-low-level mission requirements mapped to cyber assets. Through simulation of the resulting integrated model, we quantify impacts in terms of mission-based measures, for various mission and threat scenarios. Dynamic visualization of simulation runs provides deeper understanding of cyber warfare dynamics, for situational awareness in the context of simulated conflicts. We demonstrate our approach through a prototype tool that combines operational and systems views for rapid analysis.

Keywords – modeling and simulation; mission assurance; process modeling; attack graphs; cyber situational awareness

I. INTRODUCTION

In the U.S. Department of Defense (DoD) roadmap for cyber modeling & simulation (M&S), planning for integrated cyber and kinetic mission assurance is a key capability area [1]. The range of capabilities called out in the roadmap underscores the urgent need for rapid progress in this area, especially given the asymmetric nature of cyber conflict.



Figure 1. Spectrum of Cyber M&S Applications and Challenges

Of particular importance is the integration of kinetic operations with the defensive cyber operations that support them. This requires effective communication of cyber situations (and their big-picture impacts) to decision makers. In addition, there are numerous potential applications of cyber M&S, along a spectrum of increased maturity and corresponding research challenges, as shown in Figure 1.

Understanding mission resilience to cyber warfare requires bringing together layers of information from numerous sources. At the lower layers, network topology, firewall policies, intrusion detection systems, system configurations, vulnerabilities, etc., all play a part. We can combine these into a higher-level attack graph model that shows transitive paths of vulnerability. We also need to map cyber assets to mission requirements, and capture dependencies from low-level requirements to higher-level ones appropriate for decision making. Because mission requirements are highly dynamic, we need to capture time-dependent models of mission flow. Cyber attacks and defenses are similarly dynamic, and defenses generally vary depending on particular attack classes.

We introduce an approach that addresses all these aspects of mission-oriented cyber resilience, through an integrated M&S environment. This approach is called *Analyzing Mission Impacts of Cyber Actions* (AMICA). AMICA supports exploration and experimentation of the mission impacts of cyber warfare. The goal is to develop a flexible, extensible, modular, multi-layer M&S system for quantitative assessment of operational impacts of cyber attacks on mission performance. AMICA is expected to increase our understanding of dependencies between operational missions, cyber TTPs, and computing infrastructure.

II. PREVIOUS WORK

There have been numerous information-centric military exercises with aspects of mission assurance and cyber warfare. In many exercises (e.g., Global Thunder [2] and Turbo Challenge [3]), cyber security is an important component, but not the primary exercise focus. More cyber-focused exercises such as Cyber Flag [4] have integrated cyber activities with operational missions for training purposes.

M&S has been applied in more traditional military spheres, e.g., for inferring enemy intent [5], entity-based battlefield simulations [6], and command decision support [7]. However, military mission planning has yet to leverage M&S and other formal methods as part of its standard practice, especially in the area of developing cyber defensive courses of action. In short, tools such as AMICA for assessing mission impact of cyber warfare are generally unavailable for operations-level support. The defense community is aggressively accelerating cyber defense forces [8], further motivating the need for more advanced capabilities in cyber course-of-action planning.

In the cyber domain, M&S capabilities are still relatively immature. Still, previous work can be leveraged for certain components of an integrated overall M&S approach. Systems such as Topological Vulnerability Analysis (TVA) [9][10], Network Security Planning Architecture (NetSPA) [11], and NRL's ACCEPT (A Configurable Cyber Event Prioritization Tool) [12] fuse network data (topology, firewall rules, asset inventories, vulnerability scans/databases, intrusion alerts, etc.) into graph-based models for mapping vulnerability paths and prioritizing events. Capabilities such as MITRE's Cyber Command System (CyCS) [13] and Cyber Mission Impact Assessment (CMIA) [14], and AFRL's Cyber Mission Assurance [15] capture mission and cyber dependencies.

Another key enabler for cyber M&S is standardization efforts. Making Security Measurable™ [16] is a collection of standardization activities within the cyber security community. It includes Common Vulnerabilities and Exposures (CVE), Common Attack Pattern Enumeration and Classification (CAPEC), Cyber Observable Expression (CybOX), Structured Threat Information Expression (STIX), and many others. These standards cover different aspects of security data needed for building comprehensive and accurate models.

To capture the flow of mission and cyber processes, we leverage the Object Management Group (OMG) Business Process Model Notation (BPMN) [17] standard. We employ the commercial tool iGrafx [18], which extends BPMN with behavioral modeling, critical-path analysis, discrete-event simulation, Monte Carlo analysis, and experiment design.

III. APPROACH

To explore the AMICA approach, we are conducting a pilot study and developing a proof-of-concept system. We seek a flexible, extensible, modular, and multi-layer M&S environment for quantitative assessment of operational impacts of cyber attacks on specific missions, as shown in Figure 2. Thus components can be interchanged, e.g., multiple missions on an infrastructure, to support analysis of different questions.

AMICA currently includes libraries for operational (kinetic) missions, computing infrastructure on which missions depend, cyber attacker TTPs, and cyber defensive TTPs. Calibration and validation of the model occurs in concert with mission commanders, operators, and cyber defenders. In essence, we are connecting cyber effects to the kinetic domain, in the context of highly dynamic cyber warfare and mission threads. This helps commanders better maintain mission effectiveness in a force-on-force cyber-contested environment, and align defenses for best operational effect across a campaign.

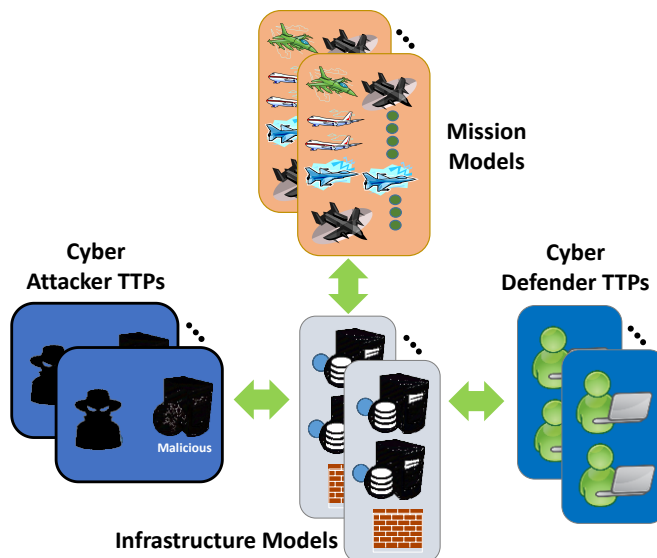


Figure 2. Modular Libraries for Model Components

For mission analysts and commanders, we seek to answer questions such as the following:

- When and where would be the most damaging attacks against the mission?
- How long before a particular attack has significant mission impact?
- How long does it take a mission to recover from an attack?
- What is more damaging to the mission: loss of reach-back availability or degradation of system assets?

For cyber defenders and analysts, we consider questions such as the following:

- What is the impact of better sensor performance, sensor location, etc.?
- How does a change to the network topology affect security posture?
- How well does the defense perform against different tiers of attacker?
- What is the impact of different defender TTPs?
- How to align workforce to cyber workload?
- What is the impact of adversary attack speed?
- What is the impact of adversary attack timing?

As illustrated in Figure 3, we employ a layered modeling structure. This allows inputs at both the operational and cyber layers to influence the behavior of the systems layer, to produce a combined effect on mission performance.

Decoupling via layers provides model independence, with shared interfaces. This enables easy migration of missions and cyber TTPs as situations dynamically evolve. Figure 3 is notional only, and does not include all the model layers actually in AMICA. For example, there are layers for mission hierarchical dependencies, cyber vulnerability dependencies (attack graphs), etc.

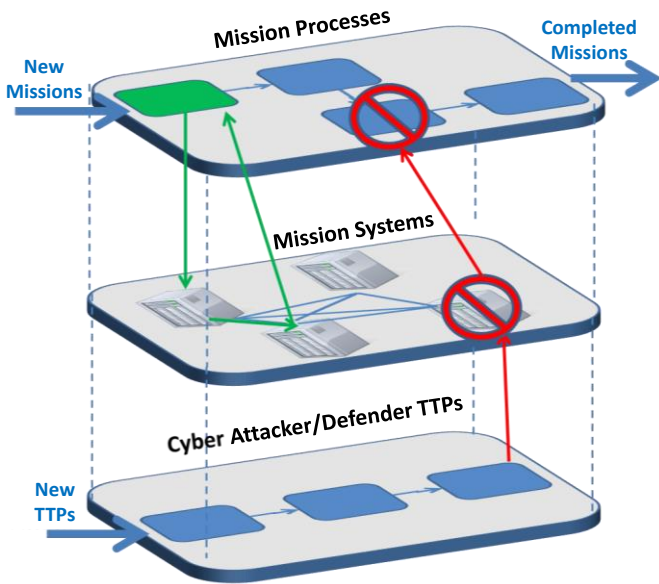


Figure 3. Model Decoupling via Layered Structure

Figure 4 shows the architectural structure of our AMICA implementation. This illustrates AMICA’s novel approach for blending workflow modeling with mission dependencies and attack graphs. Each modality (process-based and graph-based) captures a different aspect of the overall picture: workflow (process modeling) and environment (graph-based relationships, constraints, etc.). This allows workflow and environment models to be developed independently, aided by automatic generation for a given network.

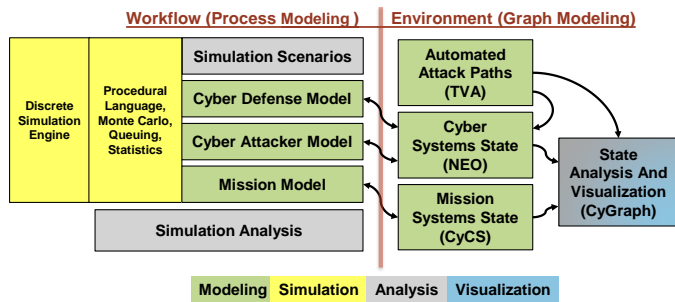


Figure 4. AMICA Architecture

Behavioral and temporal aspects of the system (workflow, timing constraints, required resources, etc.) are implemented through executable process models and stochastic discrete-event simulation (in iGrafx). Structural and functional aspects (environmental constraints, mission and system dependencies, event flows, etc.) are maintained through MIT Lincoln Laboratory’s Network Environment Oracle (NEO), and MITRE’s Cyber Command System (CyCS) [13] and CyGraph [19]. CyCS contains a directed graph comprising the information and system dependencies of each mission function. NEO contains additional topological and vulnerability information that is not captured in CyCS. CyGraph provides topological and attack graph-focused visualization of the environment and cyber attack progress. The initial state of the structural cyber (attack graph) model is generated from the

network topology, firewall rules, and system vulnerabilities via the Government Off-The-Shelf (GOTS) tool TVA [9][10]. In this way, we leverage established tools for dependency knowledge management and automated model building.

To capture workflows, decision points, workloads, resources, and temporal constraints, AMICA employs a technique called Mission-Level Modeling (MLM) [20]. MLM leverages BPMN to define, refine, and verify operational processes, decisions, and information flows among producer/consumer systems and people. It supports model libraries and parameterization to quickly assemble new prototypes. MLM handles the high degree of concurrency inherent in information-sharing operations, and explores impacts on MOEs/MOPs through simulation of mission models.

MLM is based on BPMN and discrete-event simulation, implemented in iGrafx (a commercial tool). MLM replaces static tools such as Visio and PowerPoint, providing an executable, visual model to support stakeholder collaboration to develop and validate new concepts. This provides a single model for qualitative and quantitative analysis, and enables rapid prototyping and reuse through a single modeling standard.

Figure 5 shows the operational flow among the AMICA sub-systems. The TVA tool [9][10] provides the network topology and vulnerable attack paths through the network. This represents the initial state of the network, before cyber attacks and defenses are simulated. TVA initializes NEO, which maintains dynamic cyber state under simulation and provides choices for next possible cyber states. Similarly, CyCS maintains dynamic simulation state for mission dependencies.

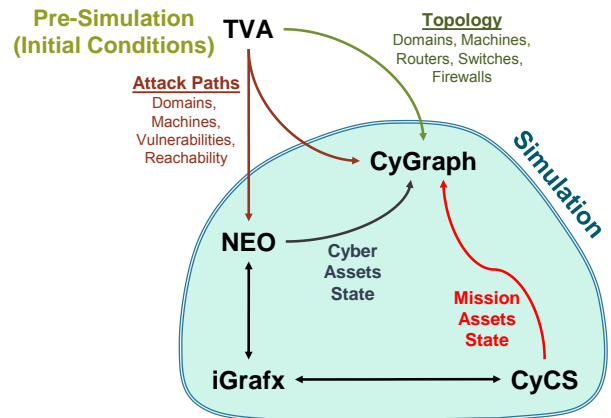


Figure 5. AMICA Operational Flow

At simulation time, iGrafx simulates mission and cyber threads concurrently, testing cyber and mission states as needed, and updating them when process tasks (i.e., cyber attacker and defender tasks) change environmental conditions. For example, when the cyber attacker process compromises a mission-critical machine, iGrafx updates the node’s state in CyCS (which propagates to higher-level mission dependencies).

Similarly, if the cyber defender process repairs the machine, its state is reset in CyCS. Asynchronously, mission tasks check the appropriate higher-level CyCS nodes upon which they depend. Throughout the entire process, CyGraph shows the dynamic state evolution through animated visualization.

IV. CASE STUDY

For our case study, we consider a key mission within a regional Air and Space Operations Center (AOC). In an AOC, an air component commander provides top-level command and control of air and space operations. In our case study, the mission focus is deliberate kinetic targeting [21], from basic target development through development and publication of the Air Tasking Order (ATO).

Thus we model, simulate, and quantitatively analyze the impact of cyber attacks on the targeting mission (number of targets successfully processed) within an AOC. Our parameterized library of AMICA modules can be rapidly reconfigured to represent different mission, cyber threat, and/or cyber defense scenarios.

Figure 6 shows the phases of progression for target development. On-going target development defines all possible targets available for strike in the area of responsibility (AOR). In preparation for an anticipated crisis, advanced target development reexamines potential targets in preparation for possible strike.

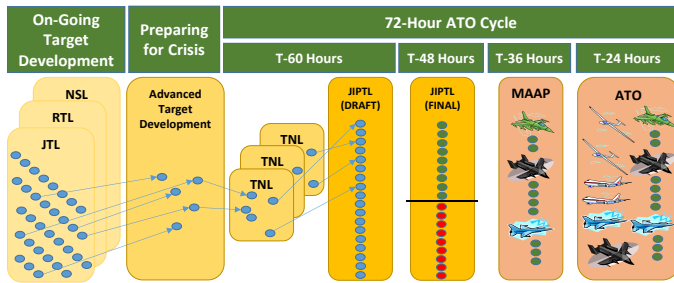


Figure 6. Target Development and ATO Process

Once hostilities actually begin, targets are nominated for potential inclusion in the ATO. Nominated targets are prioritized, and then a final target is selected based on available delivery assets. Targets are paired with assets, leading to the completed ATO.

For this case study, we leverage Mission-Level Modeling (MLM) originally developed for U.S. European Command (EUCOM) for Exercise Austere Challenge 2010 [22]. This covers the targeting process from basic target development through the Master Air Attack Plan (MAAP) and ATO, as well as Battle Damage Assessment (BDA).

This targeting model has over 200 steps, with timing and required resources per step. The model is organized as high-level modules that reference lower-level reusable library models. Figure 7 shows a high-level portion of this model.

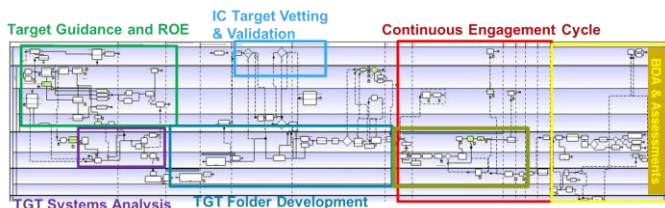


Figure 7. Portion of ATO Target Development Model

In this model, each target is tracked through the target-development process until completion, including whether the confidentiality or integrity of the target data was breached. Through simulation, we quantify mission performance and effectiveness, with metrics such as numbers of targets making each list, timing of each phase of development, workforce utilization, downtime, etc.

Figure 8 shows a high-level portion of a cyber attacker model. In this particular scenario, a phishing attack results in a malware infection, giving the adversary an initial presence inside the network. The attacker then moves laterally through the network, until a mission-critical machine is compromised. At that point, the attacker achieves the desired attack goal (compromising confidentiality, integrity, and/or availability). Depending on the scenario settings, the adversary may delay the final impact to coincide with a critical phase of the mission.

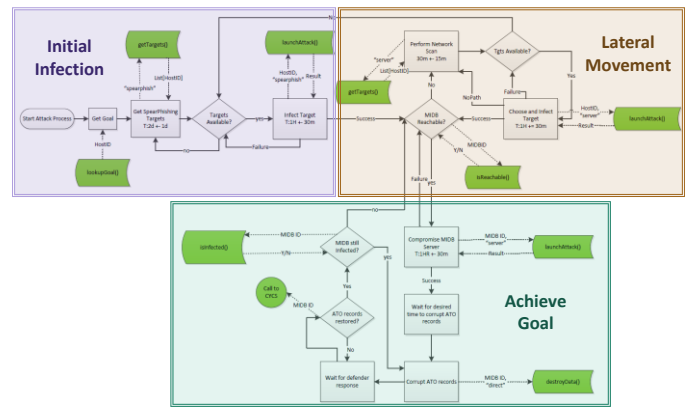


Figure 8. Portion of Cyber Attacker Model

Figure 9 shows a high-level portion of the cyber defender model. The process is triggered by an alert (intrusion detection system, user tipoff, etc.), followed by triage to understand the basic nature of the alert.

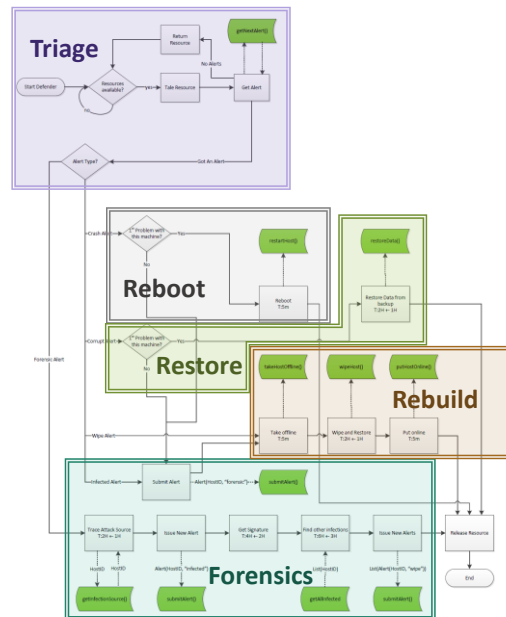


Figure 9. Portion of Cyber Defender Model

Depending on the severity of the incident and past history with the victim machine, the defender either reboots the machine, restores corrupted data, or rebuilds the machine from a non-compromised image. If an infection is detected or a machine is a victim in multiple incidents, the defender conducts more in-depth forensics. This involves searching for other infections and rebuilding victims as needed.

As for the mission model, the cyber (attacker and defender) models are modular, with higher-level models referencing sub-models. That is, process tasks (boxes) in a given model may represent entire sub-models defined elsewhere in the AMICA library.

Our cyber model leverages previous collaborative work with cyber defenders to define a process flow for their operations. This model captures adversary TTPs for major classes of attacks (email-based, browser-based, and host-based), with corresponding defensive TTPs. This collaborative work has produced a rich process diagram (in Visio), approaching 1000 steps. For AMICA, we use this as the basis for an executable model in iGrafx.

As described in Section III, the cyber attacker and defender processes (in iGrafx) interact through the Network Environment Oracle (NEO). NEO maintains state in the cyber attack graph, which the attacker and defender process models check for environmental conditions required for taking next steps (vulnerabilities, reachability, infection state, etc.).

NEO state is reflected in CyGraph [19], a MITRE tool for cyber graph analytics, interactive visualization, and animation. Figure 10 shows a representative attack graph in CyGraph, with infected machines in red and rebuilt machines in green.

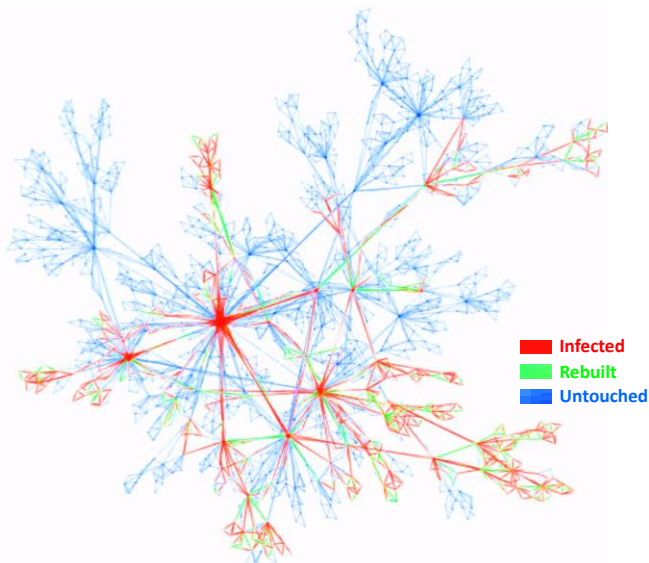


Figure 10. Cyber Attack Graph with Dynamic States

While NEO maintains state for cyber-related assets, MITRE’s Cyber Command System (CyCS) maintains state for mission-related assets. CyCS models mission dependencies as a directed acyclic graph (hierarchy). The upper levels of the hierarchy are high-level mission assets (organizations, major work products, etc.). These are mapped to subordinate entities

on which they depend. Dependencies can be conjunctive (Boolean AND) or disjunctive (Boolean OR). At the bottom of the hierarchy are those entities with no subordinates. Figure 11 shows a representative mission-dependency graph, visualized via CyGraph.

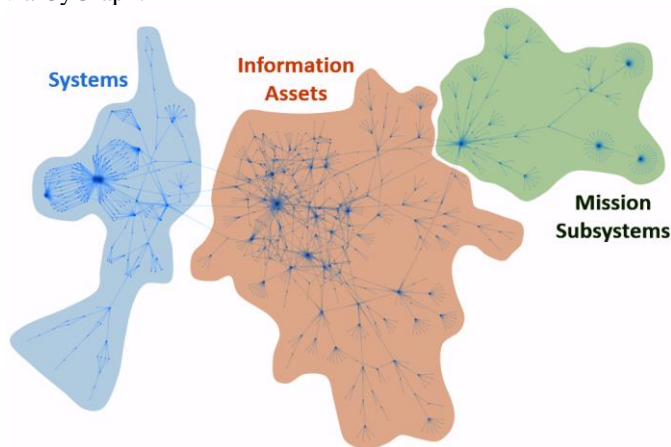


Figure 11. Graph of Mission Dependencies

As an example of the quantitative analyses available through AMICA, consider Table 1. This shows mission impact from a simulated cyber attack. In this scenario, the attack results in loss of availability of a mission-critical database service.

Table 1. Impact of Availability Attack (JTL Targets)

Cycle	Without Attacks	With Attacks	Relative Impact
4 days	9	1	88%
7 days	21	1	95%
14 days	76	70	8%

In this scenario, the attack occurs during routine operations early in the target-development process. The metric for cyber impact is a mission-based measure of performance (MOP): the number of targets that make the Joint Target List (JTL). The relative impact in the table (in percent) is then

$$Impact_{relative} = 100 \cdot [1 - (n_{with\ attacks} / n_{without\ attacks})].$$

The experiment is to determine a baseline number of JTL targets produced in the absence of an attack, and to compare that to the number of JTL targets produced when the AOC is under attack.

The results in Table 1 show a dramatic mission impact from the cyber attack. Moreover, the effects are fairly long-lasting; after a week, the relative impact is still only one JTL target produced (versus the expected 21 targets). By the end of the second week after attack, JTL target production is mostly caught up.

In these experiments, the processing of each target is simulated individually. At various points in the process, there are certain conditions, timings, etc., that have some degree of uncertainty. These are modeled as probability distributions in the appropriate points in the model. In a simulation run, Monte Carlo analysis executes the stochastic model according to model parameters.

Table 2 shows quantitative results from another AMICA simulation. This scenario is an integrity attack against a critical database during advance target development (the phase that prepares for an anticipated crisis). The mission-based MOP for measuring cyber impact is the number of targets added to the Joint Integrated Prioritized Target List (JIPTL).

Table 2. Impact of Integrity Attack (JIPTL Targets)

Cycle	Without Attacks	With Attacks	Relative Impact
4 days	574	303	47%
7 days	1098	1044	5%
14 days	1098	1087	1%

The results in Table 2 show that this attack is less impactful in terms of relative reduction in targets processed. Moreover, the AOC is able to rebound from the attack more quickly.

Figure 12 shows the relative impact on mission performance for the two attack scenarios: (1) availability attack against producing the JTL in routine early development, and (2) integrity attack against producing JIPTL in advanced target development in preparation for crisis.

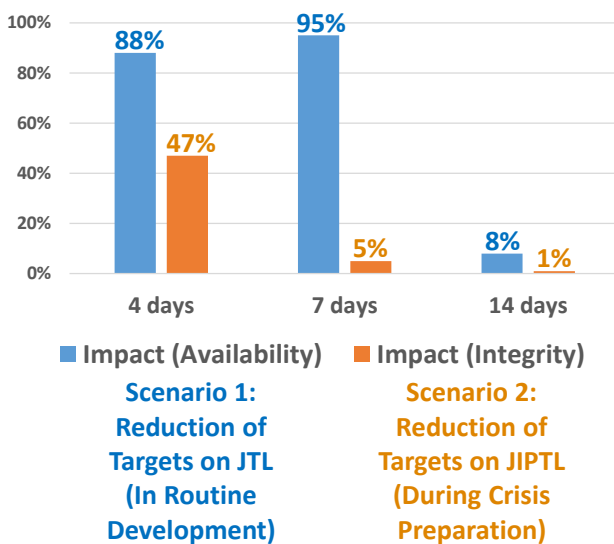


Figure 12. Relative Impact for Two Attack Scenarios

Of course, not all target-production numbers may be equally important. For example, the criticality of the development phase itself may be a strong factor in overall impact. But it is clear that AMICA provides a quantitative approach to address these kinds of questions, based on simulation of vetted models for missions and cyber TTPs.

We are investigating a range of more advanced attacks against different portions of the targeting process, such as data alterations that interfere with battle damage assessment, move target locations, inject discrepancies that force massive rework, etc.

V. SUMMARY AND NEXT STEPS

We have described an integrated approach for quantitative analysis of mission impact from cyber attacks, known as AMICA (Analyzing Mission Impacts of Cyber Actions). AMICA defines process models for mission threads and cyber tactics, techniques, and procedures (TTPs). These process models are designed as a hierarchically-decomposed library of reusable modules, for rapid reconfiguration and prototyping.

AMICA process models are probabilistic and executable, supported by discrete-event simulation and stochastic Monte Carlo analysis. Through simulation of mission and cyber models, we are able to quantitatively assess mission impact from cyber attacks. Monte Carlo analysis provides distributions over multiple simulation runs, for bounding uncertainty in results. For process modeling and simulation we apply industry-standard Business Process Modeling Notation (BPMN) implemented in a commercial tool (iGrafx).

While process models capture workflow and behavioral phenomena, processes necessarily operate within the structural constraints and dependencies of a particular environment. This includes dependencies between mission requirements and cyber assets, as well as constraints on attacker freedom of movement. We capture these through graph models (mission-dependency graphs and attack graphs), which are dynamically updated under process-model simulation.

This novel merging of M&S modalities supports dynamic simulation while leveraging established tools for cyber/mission knowledge management and automatic model building (e.g., attack graphs). Through simulation of this integrated multi-modal model, AMICA quantifies cyber impacts in terms of mission-based measures, for desired mission and threat scenarios. We provide animated visualizations of simulation runs, showing environmental state changes during the interplay of cyber force-on-force warfare.

We demonstrate AMICA through a case study, showing cyber impacts against a particular kinetic mission: targeting for Air Tasking Order (ATO) development in an Air and Space Operations Center (AOC). We model, simulate, and quantify the impact of cyber attacks on the targeting mission. We show impact results for two attack scenarios (availability and confidentiality) against different phases of the target-development process. Our simulations quantify cyber impact in terms of mission-relevant measures (numbers of targets completed) over time.

In the future, we plan to develop a more rigorous experimental framework for posing hypotheses, designing experiments, and validating results. The goal is to provide a rich and agile environment for gaining scientific insights. Examples of such hypotheses include:

- *Levels of Fidelity:* Given a threat model, what is the right level of fidelity to predict mission impact with sufficient accuracy?
- *Threat Classes:* For a given set of threat classes, what level of coverage is sufficient to maintain mission readiness?

- *Attacker TTPs*: What is the right degree of automation to achieve a desired mission impact? How much knowledge is required for a desired impact?
- *Attack Dynamics*: When should the adversary attack to have the highest mission impact? Which attack mode (e.g. fast smash-and-grab or slow-and-stealthy) can cause greater mission impact? How many concurrent attacks can the mission withstand?
- *Defense TTPs*: Under what conditions are static defenses inadequate? What is the best combination of static, dynamic, and synergistic defenses?
- *Attack Surface and Resiliency*: What degree of diversity gives adequate protection against zero-day attacks? What is the right balance between diversity, redundancy, containment, and cost?

Overall, AMICA merges cyber and kinetic domains (mission threads, cyber TTPs, network environment, etc.) into a common M&S environment, with complementary modeling modalities (process-based and graph-based). This provides a strong foundation for answering these kinds of questions.

ACKNOWLEDGMENTS

We would like to acknowledge the assistance provided by members of United States European and Pacific Commands, and the 603rd and 613th Air & Space Operations Centers, as well as Scott Foote of the MITRE Corporation. This work was performed in support of Dr. Steven King from Information Systems & Cyber Technology, Office of the Assistant Secretary of Defense (Research & Engineering) under contract W56KGU-14-C-0010. MIT Lincoln Laboratory work was performed under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

REFERENCES

- [1] Steven King, "Defense Cyber S&T Strategies & Initiatives," DoD/DHS Small Business Innovation Research Workshop, <https://www.dhs.gov/sites/default/files/publications/csd-sbir-2013-drsteven-king.pdf>, 2013.
- [2] U.S. Strategic Command Public Affairs, *Global Strike Forces Participate in USSTRATCOM Command, Control Exercise*, web page, <http://www.afgsc.af.mil/news/story.asp?id=123429750>.
- [3] GlobalSecurity.org, *Turbo Challenge*, web page, <http://www.globalsecurity.org/military/ops/turbo-challenge.htm>.
- [4] U.S. Cyber Command, 'Cyber Flag' Exercise Tests Mission Skills, web page, <http://www.defense.gov/news/newsarticle.aspx?id=123621>.
- [5] Alexander Kott, Michael Ownby, "Tools for Real-Time Anticipation of Enemy Actions in Tactical Ground Operations," 10th International Command and Control Research and Technology Symposium, 2005.
- [6] Robert Whittman, and Cynthia Harrison, "OneSAF: A Product Line Approach to Simulation Development" European Simulation Interoperability Workshop, 2001.
- [7] John Surdu, Kevin Kittka, "The Deep Green Concept," Spring Simulation Multiconference, 2008.
- [8] Lieutenant General Edward Cardon, statement before U.S. House of Representatives (Armed Services Committee), March 4, 2015.
- [9] Sushil Jajodia, Steven Noel, Brian O'Berry, "Topological Analysis of Network Attack Vulnerability," in *Managing Cyber Threats: Issues, Approaches and Challenges*, Springer, 2005.
- [10] Sushil Jajodia, Steven Noel, "Topological Vulnerability Analysis," in *Cyber Situational Awareness, Advances in Information Security 46*, Springer, 2010.
- [11] Michael Artz, *NetSPA: A Network Security Planning Architecture*, Masters thesis, Massachusetts Institute of Technology, 2002.
- [12] Anya Kim, Myong Kang, Jim Luo, Alex Velazquez, *A Framework for Event Prioritization in Cyber Network Defense*, Technical Report NRL/MR/5540--14-9541, Naval Research Laboratory, <http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA608707>, 2014.
- [13] The MITRE Corporation, *Cyber Command System (CyCS)*, web page, <http://www.mitre.org/research/technology-transfer/technology-licensing/cyber-command-system-cycs>.
- [14] Scott Musman, Aaron Temin, Mike Tanner, Dick Fox, and Brian Pridemore, *Evaluating the Impact of Cyber Attacks on Missions*, 5th International Conference on Information Warfare and Security, 2010.
- [15] Air Force Research Laboratory (AFRL), *Cyber Mission Assurance*, white paper, <http://www.wpafb.af.mil/shared/media/document/AFD-110516-046.pdf>, 2011.
- [16] The MITRE Corporation, *Making Security Measurable™*, web page, <http://makingsecuritymeasurable.mitre.org/>.
- [17] Object Management Group, *Business Process Model and Notation*, web page, <http://www.bpmn.org/>.
- [18] iGrafx, *Process Modeling – Communicate Business Processes Clearly, Completely and Efficiently*, web page, <http://www.igrafx.com/solutions/business-challenges/process-modeling>.
- [19] Steven Noel, Eric Harley, Kam Him Tam, Greg Gyor, "Big-Data Architecture for Cyber Attack Graphs: Representing Security Relationships in NoSQL Graph Databases," IEEE Symposium on Technologies for Homeland Security (HST), 2015.
- [20] The MITRE Corporation, *Systems Engineering Guide – Collected Wisdom from MITRE's Systems Engineering Experts*, technical paper, 2014.
- [21] *Annex 3-60 Targeting – U.S. Air Force Doctrine*, Curtis E. LeMay Center for Doctrine Development and Education, training manual, <https://doctrine.af.mil/download.jsp?filename=3-60-Annex-TARGETING.pdf>.
- [22] Capt. Brendan Simison, Massachusetts Air National Guard 102nd Air Operations Group, *102nd Air Operations Group Participates in AUSTERE CHALLENGE - 10*, web page, <http://www.102iw.ang.af.mil/news/story.asp?id=123205843>.

Modeling Risk and Agility Interaction on Tactical Edge

James R. Morris-King and Hasan Cam

Abstract— Cyber-risk models often explore exploitation methods, agility maneuvers, and mitigation techniques to reduce vulnerabilities/counter risks. Cyber-agility models employ quarantine and inoculation-like maneuver procedures to protect vulnerable systems from a known, detected threat. Although fairly effective, these procedures often diminish network function in tactical environments which adversely impact mission assurance and increases system damage beyond the exploit itself. This paper proposes a novel risk-classifier model which assesses influence to ensure tactical edge networks function during an attack by preserving critical, high-risk nodes. Unlike other risk assessment strategy models, our model employs temporal propagation graphs to capture the impact of vulnerability exploits. These high-risk nodes are supported by an agility process that reacts to an attack by quarantining exploited systems and designating viable successors to carry on key mission functions with varying degrees of service availability. We validate this model via an agent-based simulation. Our simulation results indicate that risk analysis-supported agility maneuvers outperform reflexive strategies.

Index Terms—tactical edge network, risk, agility, agent-based simulation, ecological modeling, epidemic system, risk propagation

I. INTRODUCTION

Within the tactical edge paradigm, *battlespace agility* is a warfighting concept defined as the speed at which the warfighting organization is able to transform knowledge into actions for desired effects in a battlespace (Libicki & Johnson, 1996; Mitchell, 2012a). The deterministic nature of cyber-physical battlespace creates asymmetry in cyber warfare at the tactical edge. This determinism allows adversaries to plan, coordinate and launch attacks effectively, while defenders lack the capabilities to predict attack strategies or react in a timely fashion (Mitchell, 2012b). The growing need to respond quickly to cyber threats in the modern battlespace presents many challenges to operators concerned with preserving integrity and mission-assurance of cyber Command & Control (C2) systems. One challenge that this paper addresses is how to minimize the adverse impact of vulnerability exploitations in critical nodes by employing network influence assessment

to choose the best practical agility maneuver.

Network influence is a concept drawn from social networking theory and is defined in Kempe et al. (2003) as the extent to which individuals are likely to be affected by decisions of their neighbors. In the tactical edge, influence is considered as a measure of one cyber-physical system's (CPS) ability to control the behavior and state of other systems. In this context, cyber agility is a reasoned modification to a CPS in response to a functional, performance, or security need (McDaniel et al., 2014). Perhaps the greatest need for agility can be found in formulating *agility maneuvers*, or strategies to mitigate damage to cyber networks when CPS are compromised by an attacker. There exists a suite of known metrics for evaluating the scope and effective impacts of cyber agility maneuvers (Pfister, 2012). These metrics include robustness, resilience, responsiveness, and adaptation measures which are used for impact assessment rather than future decision-making. Traditional models of cyber agility impact assessment do not consider properties such as mobility, temporality, and environmental interference when evaluating threats to CPS (Riley & Ammar, 2002). This makes the development of predictive analytical models difficult, and represents an open challenge in cyber risk assessment (CRA).

Tactical edge networks share many properties in common with traditional mobile ad-hoc networks (MANET), which are defined by Burbank et al. (2006) as “deployed networks supporting users and platforms within the tactical operations region”. They are often sparse and dynamical, consisting of a heterogeneous mixture of various autonomous and human-operated networked systems. When attackers penetrate these networks, it is the role of cyber operators and network specialists to devise and execute countermeasures, or agility maneuvers, to mitigate these attacks and recover damaged or compromised systems. An agility maneuver may operate in any of the domains available to socio-technical security models (physical, virtual, cognitive, and policy). Regardless of the selected domain, the purpose of the agility maneuver remains; altering the vulnerability landscape such that the risk of present and future attacks are degraded or eliminated.

In order to properly relate risk and agility, it is necessary to develop cyber-social models that are able to classify risk controllers in the cyber environment and demonstrate the effect of agility maneuvers on mission-critical vulnerabilities. To that end, we propose a risk-classifier model that

J. R. Morris-King is with Army Research Lab, Adelphi, MD 20773 USA (e-mail: james.r.morris-king-ctr@mail.mil).

H. Cam, Jr., is with Army Research Lab, Adelphi, MD 20773 USA (e-mail: hasan.cam.civ@mail.mil).

incorporates mission-assurance evaluation, criticality, and propagating risk analysis. Our contributions are multifold. We propose an ecology-inspired influence metric and a cost function to approximate the impact to the network of a propagating cyber-attack. This metric is then used to support an agility maneuver which selects successor host nodes with minimal risk. We include the physical and temporal natures of cyber risk on the tactical edge by incorporating system mobility as well as incorporating power and bandwidth constraints in the model. The response of the network to propagating exploit is measured in terms of cascading damage to mission-critical nodes and the cost of mitigation and recovery operations. We then validate the model through the development of a multi-agent simulation.

The remainder of this paper is organized as follows. Relevant work is discussed in Section II. Section III presents our risk model for tactical edge. Section IV provides how risk influence can be used to drive optimal agility maneuver selection. Section V describes simulation model and results. Concluding remarks are made in Section VI.

II. RELEVANT WORK

Mattson (2007) highlighted the need for new cyber models which included the impact of the use mobile devices have on network security. Libicki (2007) proposed the notion that three distinct layers must be represented in models of cyber systems the physical layer, the syntactic layer, and the semantic layer. These layers possess distinct attributes which allow models to display meaningful interactions. Shapiro & Varian (1999) note the complementary network effects of adding or removing highly critical network nodes in distributed systems. Fortson (2007) highlights a number of deficiencies of common CRA practices and highlights various objectives for impact analysis which include: documentation of dependency relationships; ability to show effects of timing and duration of attacks on cyber targets, and prediction of mission-impact. A wide variety of reward-based system dependability and performance measures are discussed in Sanders & Meyers (1991) and Trivedi (2001). Various proactive mitigation maneuvers were explored by Haadi et al (2014) who proposed a novel moving-target-defense strategy which was evaluated via deterrence, deception, and detectability metrics. Whiteman (2008) and others have proposed tools for performing CRA which leverage simulation and automated mission-plan validation. However these models have little use in predicting multi-stage propagating exploits (Yu, 2013).

III. PROBLEM STATEMENT

We consider a mobile network as a tactical edge that has n mobile nodes, each corresponding to a vehicle and/or user with devices. Each device can have one or more assets (e.g., software, hardware, data, service). This network is represented as a directed graph, denoted by $G = (V, E)$, where V is the set of n vertices, and $E \subseteq V * V$ is the set of all directed edges representing the connections between mobile nodes. A directed edge (i, j) from node i to node j exists iff node i can

transmit to node j directly. When an asset of node k is infected, exploited, or suffers a fault, this failure has the potential to influence all of k 's neighbors, based on their individual susceptibility. The influence exerted on node j from an exploit or fault at node i is expressed via an influence metric ψ_{ij} , where $\psi_{ij} = [0, 1]$. The directed graph of influence is denoted as $IG = [\psi]$.

The local composite of directed influence is described by the variables ρ_k and τ_k , where ρ_k represents the sum of ψ on all outgoing edges from node k in IG , and τ_k represents the sum of ψ on all incoming edges terminating at node k in IG . If $\rho_k > \tau_k$, then node k is said to be a *controller* node in the network graph, else if $\rho_k < \tau_k$ then node k is said to be a *dependent* node in the network graph.

Criticality is assessed as a function of the number of shared assets on node k , the relative importance of those assets to the operation, the nodes which access these assets, and the communication paths which utilize node k as a hub or sink. It can be represented by the following equation:

$$\gamma_k = (A_k * S) + DN_k + ICP_k \quad (1)$$

where A_k represents the available assets at node k , S is a scalar modifier representing the importance of those assets, DN_k represents the number of nodes which treat node k as host, and ICP_k represents the communication paths which pass through or terminate at node k . From this we say that there exists a set of critical nodes B such that $B \subseteq G$. A node $k \in B$ iff $\gamma_k > \eta$, where η is an arbitrary criticality threshold determined at the start of an operation. The set of nodes in B may change over time. This variability is subject to network topology changes resulting from node mobility, node loss via malicious exploit or power depletion, and/or cyber agility maneuver by network managers.

Criticality also functions as a component of the impact measure I_k , which indicates the suffered by the network if this node is lost. This measure is expressed as:

$$I_k = \frac{\gamma_G - \gamma_k}{\gamma_G} \quad (2)$$

where I_k represents the damage to the system from the loss of node k and γ_G is the criticality of the network. γ_G is expressed as:

$$\gamma_G = (\sum_{i \leq n}^i \gamma_i) \quad (3)$$

where γ_k is the criticality of node k , n is the number of active nodes in the network, and γ_i is the criticality of the i th node.

Performing agility maneuvers on network nodes engenders a *cost* measured by C , which is expressed by the following equation:

$$C = D * PR * ET \quad (4)$$

where D represents the relative sophistication or expertise required to perform the maneuver, PR represents the financial investment (in terms of parts and labor) of performing the maneuver, and ET represents an estimate of time required to complete the maneuver.

Using these measures we are able address the following problems:

Problem 1: Assess influence and propagation of vulnerability exploitation in tactical edge by incorporating its main features (mobility, power, bandwidth, delay, etc.) and identify high-risk nodes based on their criticality.

Problem 2: Characterize a set of basic cyber agility maneuvers based on common network security practices with respect to I and C . Show via simulation how these maneuvers can be deployed under cost and resource-constrained scenarios using influence as a classifier heuristic to minimize total network impact (I).

IV. INFLUENCE MODEL FOR TACTICAL EDGE

The influence model for the tactical edge environment is formed from the union of two sets of operational requirements; mission assurance and criticality.

A. Mission Assurance

To represent mission assurance across nodes, we modify the Mission-Service-User (MSU) model proposed by D'Amico et al. (2010) and Cam and Mouallem (2012). In this view, cyber operations have four critical components:

- 1) *Basic assets, such as data stores, routers, networked sensors, etc.*
- 2) *Services rely on assets to deliver information or capabilities across the network*
- 3) *Tasks rely on time-sensitive service availability to accomplish mission-critical activities*
- 4) *Missions are composed as a series of interrelated tasks organized to accomplish some tactical goal.*

These four components form the Mission-Task-Service-Asset model. The integrity of these four control regimes mimics the hierarchal interdependence of a natural ecosystem, where the provisioning of sensitive ecological services is predicated on a complex web of heterogeneous species and environmental interactions. Like its biological counterpart, the mission ecosystem is vulnerable to cascading disruption at each level of control.

From the purview of assurance, mission success is predicated on the timely completion of mission-critical tasks. These tasks rely on service availability, which in turn relies on asset availability. Assets and services are maintained and delivered by network nodes, whose accessibility may vary based on endogenous (internal state) or exogenous (environmental) factors. Because a service on one node may require assets held by another node, service availability is as much a function of network topology as it is individual system integrity.

B. Criticality

Our conceptual model of node risk relies on an understanding of node influence ρ_k and τ_k , which are derived from a node's asset and service relationships with the broader network and as well as its topology of the graph. More formally, influence in the tactical network ecosystem is a

composite function of a node's local neighborhood (in-degree, out-degree, and betweenness), the nodes across the network that rely on services it provides, and its status as a hub for communication between non-adjacent nodes. For example, a vehicle-mounted network server which provides services to nearby warfighters is tasked with maintaining a minimum number of active connections in order to satisfy some time-sensitive task (such providing access to mission-specific intelligence or web-servers). Failure of the network server not only results in the loss of function of that device but also the loss of function of any CPS relying on it to connect to other devices. Additionally, any mission with a requirement that access to that node be maintained may be delayed or compromised due to the time and cost of recovery. Thus, the network server node exerts influence not just in the network topology, but also in the physical, social, and mission-assurance domains. We capture this influence as follows:

- Let q be the connected neighbors of node k , where an edge between two nodes is determined to exist if the source node is a member of the subset of local neighbors using k as a hub.
- Let r be the number of unique nodes that rely on services from node k .
- If $q < r$, then $\gamma_k = |q - r|$
- Else if $q \geq r$, then $\gamma_k = -1 * |q - r|$

While this relation is useful, it does not distinguish between nodes whose criticality is dictated by centrality. Centrality concepts were first developed in social network analysis and are used to identify the most influential hubs in social networks and key infrastructure nodes in the Internet or urban networks. In viral outbreak models this measure is used to identify the main spreaders of infectious disease in a population. There are several common measures for determining degree centrality in networks which have particular affordance for our problem, determining important nodes in a tactical edge environment. For the purpose of simplicity, we focus on betweenness as the centrality component of our criticality metric.

Betweenness centrality quantifies the number of times a node acts as a bridge along the shortest path between two other nodes. It was introduced as a measure for quantifying the control of a human on the communication between other humans in a social network by Freeman (1977); however, in our model, we view betweenness as a marker for whether the node in question behaves as a risk controller in the battlefield network. The formula for determining betweenness of node v in graph $G: = (V, E)$ with V vertices follows:

$$W_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}} \quad (5)$$

where σ_{st} is total number of shortest paths from node s to node t and $\sigma_{st}(v)$ is the number of those paths that pass through v . Of course, this formulation rests on the assumption that routing between nodes follows a shortest-path strategy. Other routing strategies or external constraints might make Brandes' variation of the algorithm more suitable, because it corrects for edges being counted multiple times (Brandes, 2001).

V. TEMPORAL RISK EVALUATION

A. Risk Model

Risk magnitude, R_i , is the measure of risk entering an ecosystem compartment i at a given time interval t . This risk can be self-generated (endogenous), via such conditions such as equipment failure, software error, and accidental misuse by a user, or it may arise from external sources such as other compartments (nodes) and the environment itself (exogenous). Risk magnitude is further separated into three parameters: risk intensity (RI_x), probability of risk occurrence (P_x), and compartment sensitivity (S_i). Together, these parameters determine the input risk value (R_i) at a given node as follows:

$$R_i = RI_x * P_x * S_i, 0 \leq R_i \leq 1 \quad (6)$$

Where RI_x refers to the risk intensity resulting from a state change caused by the exploitation of vulnerability x , P_x refers to the probability of that exploit occurring, and S_i is a constant representing the degree of sensitivity of compartment i . Taken together, risk magnitude becomes useful shorthand for identifying dominant compartments in tactical networks.

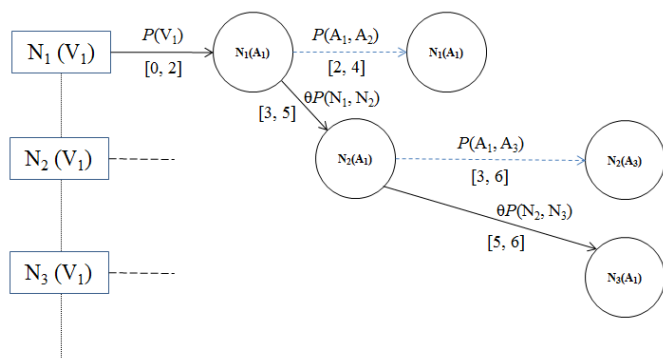


Fig. 1. VTG model ($t=10$) of a three-node system with a single shared asset vulnerability, V_1 at A_1 .

To better understand the temporal nature of vulnerability exploitation in networked systems we developed a vulnerability timing model for tactical network nodes based on hybrid failure propagation modeling. The resultant vulnerability timing graphs (VTG) illustrate the temporal and probabilistic nature of system vulnerabilities that propagate between nodes. The VTG illustrates the timing window of events, $r = [V_{min}, V_{max}]$, as well the probability of propagation, $P(N_i, N_j)$.

B. Risk Mitigation

Minimal risk optimization strategies are procedures that move the entire system towards the most secure state possible with least risk overall. In traditional cyber-security analysis, the system is modeled using two states, i.e., secure and insecure (compromised). Actions which move the system from a state of high-risk to a state of lower-risk while preserving function are known as minimal-risk maneuvers. Selecting actions which consistently perform this transition is hard in the presence of uncertain information and random processes.

Suspending or terminating a service component is oftentimes desirable if it protects the larger system, but it is harmful in response to a false alarm. Deliberate triggering by a malicious adversary might also cause self-inflicted denial-of-service.

In a control-theoretic model, the system consists of two features: (1) a discrete-time dynamic system and (2) a cost function that is additive over time. The cost function is additive in the sense that the cost incurred accumulates over time. However, because of the presence of uncertainty in the state, the cost is generally presented as a random variable which cannot be meaningfully optimized (Rowe et al., 2013). While optimizing cost may be difficult, it is possible to calculate maneuver costs by incorporating changes in risk between system states.

As indicated earlier, we assess temporal risk in tactical networks by computing R for each node in the network at each timestep. This can be combined with the impact measure I_k and the criticality score γ to identify vulnerable nodes evaluate the functional cost of a mitigation maneuver after a node is compromised. The network evaluation operation can be summarized as follows:

1. Scan graph G for in infected nodes
2. If a node is infected, add it to the set of nodes pending treatment, *INFECTED*.
3. If a node is not infected, add it to the set of susceptible nodes, *SUSCEPTIBLE*
4. Calculate R and γ for each node in the graph

Using these observations we formulate an agility maneuver with the goal of mitigating future risk to the network via a replacement operation. We set an arbitrary threshold η such that any node with $\gamma > \eta$ is considered ‘critical’. From *INFECTED*, select the critical node with the greatest I and add it to the set *QUARANTINED*. Then select a new node from *SUSCEPTIBLE* as a successor iff it is eligible. Nodes are considered eligible for succession if they are able to provide similar capabilities to those which were provided by the quarantined node. Depending on the constraints, this process may be repeated until *INFECTED* is empty, time elapses, and/or some finite resource measure is exhausted.

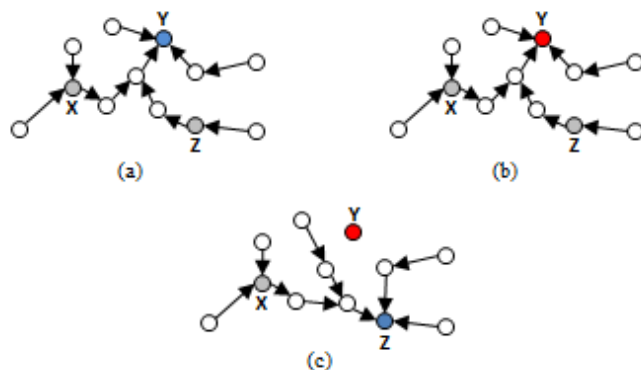


Fig. 2. Selecting a new critical node after infection. (a) pre-exploit: nodes X , Y , and Z are critical nodes with Y being selected and active. (b) An exploit occurs on node Y , leading to a quarantine operation. (c) post-exploit: node Z retains the least risk and is selected to replace Y .

Consider the scenario represented above (fig. 1) on graph G where X , Y , and Z are nodes which provide mission-critical services to the network. Y is initially selected as the critical node due to its higher C -value. Nodes X , Y , and Z provide similar services across the network. In this example, let $R_Y < R_X < R_Z$ where R_n represents the composite risk at node n . In this example, after the loss of Y , X is selected as the new source node.

This particular maneuver mimics the function of real-life ecosystems, which often adjust to the loss of ecological compartments, services, or species by exposing or promoting alternative ecological niches which can serve the remaining population in a similar fashion. From the purview of cost analysis, this maneuver is considered a ‘sealing off’ or quarantine function where the security of the node is unchanged while its capability is reduced.

C. Recovery

Restoring function to lost nodes may also follow the same process, where critical nodes under quarantine may be evaluated for healing operations (e.g. self-healing, patching, etc.) before being allowed to rejoin the network. Consider the cost function discussed in (4), risk magnitude in (6), and impact score in (2). We can inject these measures risk analysis as follows:

$$p_n(x) = \frac{R_n * I_n}{C_x} \quad (6)$$

Where x is an agility maneuver performed on node n , R represents the magnitude of risk at node n , I represents the impact of losing node n , and C represents the cost of performing agility maneuver x . This formula allows us to interpret the relationship between various operations with respect to their cost. For example, the trade-off between quarantine, self-healing, and patching can be represented with the following relationship:

$$p_n(\text{quarantine}) \leq p_n(\text{selfhealing}) \leq p_n(\text{patching})$$

This formula can be further refined to include terms covering the various cost-modifiers and constraints of network operations such as power, bandwidth, operator training, infrastructure repair, and protocol & policy development.

VI. SIMULATION & EXPERIMENTATION

In this section, we give some preliminary results for influence and cohesion scores for both mobile and non-mobile experimental models. These results are intended to illustrate the capabilities of our MANET model to accurately replicate the behavior of propagating attacks on tactical nodes, and not intended to be a comprehensive study of vulnerability exploitation on such systems. The values in Table 2 show the effect of mobility on risk-force, risk-intensity, and network cohesion.

TABLE I
TANVIS EXPERIMENTAL RESULTS (BY GROUP)

Experiment Group	Risk Force	Intensity	Cohesion
Stationary, No Exploit	0.0	0.0	1.0
Stationary, Exploit	0.71311	13.4146	0.35102
Mobile, No Exploit	0.0	0.0	0.44439
Mobile, Exploit	0.014	8.2112	02.1718

A. Scenario

Consider a propagating exploit where a set of infected nodes $I \subseteq G$ are the subject of a propagating exploit at time $t = 0$. Observation of infected nodes by network operators may not be immediate, and is controlled by an observation probability OP_x which scales with respect to length of infection. Network operators are cost constrained and may only spend an arbitrary amount of their budget to respond to observed exploits. Let $B \subseteq G$ be the set of critical nodes. At each timestep t , calculate the risk graph RF and label controller and dependent nodes. For all observed nodes, select k where $k \in I$ AND B , k maximizes ρ , and minimizes C . If $C(k) < budget$

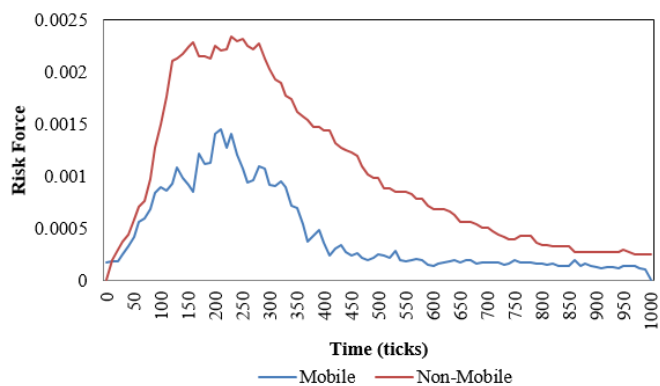


Fig. 3. Composite network risk (R) across mobile and non-mobile network models during a propagating worm attack.

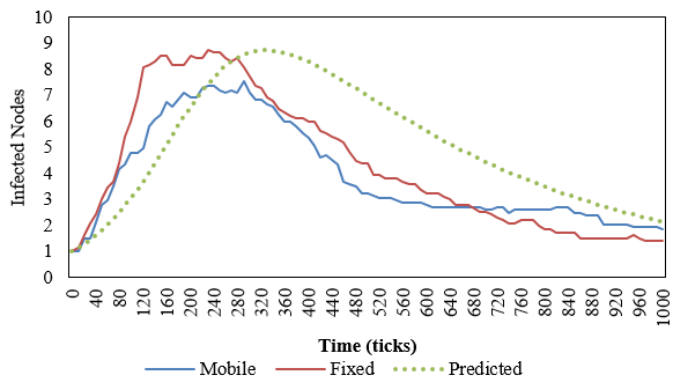


Fig. 4. Worm exploit simulation with SIR prediction.

Total risk force (Fig. 3) follows a predictable pattern with respect to change in infected population predicted by the Standard Epidemic Model, with peak vulnerability occurring in network topologies that with high levels of cohesion. This is understandable, as sparse networks create compartments which are isolated from exploit due to distance or complete

inaccessibility. This phenomenon is common to topologies displaying small-world characteristics (most nodes are not neighbors of one another, but most nodes can be reached from every other by a small number of hops or steps) and low cohesion. The reduced risk force and intensity indicate a sort of topological resistance which results node mobility. Likewise, average risk intensity was lower in mobile networks as nodes dispersed prior to contact with infected neighbors. The average subnetwork size for Random Waypoint was 4.6211 (compared to 12 in the fixed network), which is in keeping with network cohesion. This can be explained was lowest in mobile network suffering from worm attack due to the combined loss of node-connectivity from mobility and exploit.

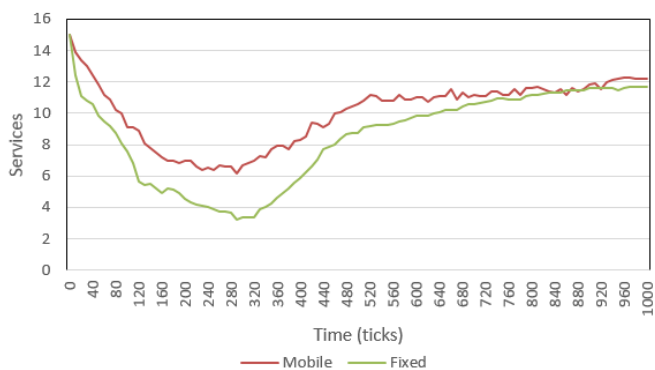


Fig. 5. Average service availability in mobile and fixed network models.

VII. CONCLUSION

In this work we present a design and implementation of a model for risk analysis of agility maneuvers in a simulated tactical network environment. This model was evaluated via agent-based simulation of a theoretical tactical environment including mobile warfighters, their attendant digital devices, and an automated network analysis module. Findings from preliminary experimentation indicate that risk-based agility maneuvers such as quarantine and patching operations increase mission assurance by maintaining network function even in scenarios where battery power and bandwidth limit the ability of network operators to reach every vulnerable node. Possible future improvements include: the development intelligent mobility models, alternative asset distribution across nodes, advanced behavior models for individual warfighter agents. We intend to extend these simulation models with data drawn real tactical network for the purpose of cross-validation.

REFERENCES

Brandes, U. (2001). "A faster algorithm for betweenness centrality" (PDF). *Journal of Mathematical Sociology* 25: 163–177.

Breslau, L., Estrin, D., Fall, K., Floyd, S., Heidemann, J., Helmy, A., Huang, P., McCanne, S., Varadhan, K., Xu, Y., Yu, H. 200. Advances in network simulation, *IEEE Computer* 33 (5) 59–67.

Burbank, Jack L., Philip F. Chimento, Brian K. Haberman, and W. Kasch. "Key challenges of military tactical networking and the elusive promise of

MANET technology." *Communications Magazine, IEEE* 44, no. 11 (2006): 39-45.

Cebrowski and Garstka, *Network-Centric Warfare: Its Origin and Future*, 28.

Chen, Wei, Yajun Wang, and Siyu Yang. 2009. "Efficient influence maximization in social networks." *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM

Command, US Joint Forces. "Commander's Handbook for Attack the Network." (2011).

Fortson L. (2007). "Towards the Development of a Defensive Cyber Damage and Mission Impact Methodology", AFIT Masters Thesis, March 2007

Freeman, L. 1977 "A set of measures of centrality based upon betweenness". *Sociometry* 40: 35–41

Grimm, V. 2005. "Pattern-oriented modeling of agent-based complex systems: lessons from ecology." *science* 310.5750 (2005): 987-991.

Goyal, A., Bonchi, F., and Lakshmanan, L. V. S. 2010. "Learning influence probabilities in social networks." *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 2010.

Keeling, M. J., and Eames, K. T. D. 2005. "Networks and epidemic models." *Journal of the Royal Society Interface* 2.4 (2005): 295-307.

Kempe, David, Jon Kleinberg, and Éva Tardos. "Maximizing the spread of influence through a social network." In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 137-146. ACM, 2003.

Kempe, David, Jon Kleinberg, and Éva Tardos. "Influential nodes in a diffusion model for social networks." *Automata, languages and programming*. Springer Berlin Heidelberg, 2005. 1127-1138.

Libicki, M. C. & Johnson, S. E. eds. (1996) "Dominant Battlespace Knowledge". April 1996.

Libicki, M. C. *Conquest in Cyberspace: National Security and Information Warfare*, (Cambridge MA: Cambridge University Press, 2007), 8-9.

Mattson, Jeffrey A. "Cyber Defense Exercise: A Service Provider Model." In *Fifth World Conference on Information Security Education*, pp. 81-86. Springer US, 2007.

Mitchell, W. (2012a) *Kitae I: Battlespace Agility in Helmand: Network vs. Hierarchy C2*. (Copenhagen: Royal Danish Defence College Press, 2012)

Mitchell, William. (2012b) *Three C2 Models for Military Agility in the 21st Century*. (Copenhagen: Royal Danish Defence College Press, 2012)

Moore, David., C. Shannon, J. Brown, "Code-Red: a Case Study On the Spread and Victims of an Internet Worm," *Proceedings of the Internet Measurement Workshop*, 2002.

Mumby, Peter J., et al. "Ecological resilience, robustness and vulnerability: how do these concepts benefit ecosystem management?." *Current Opinion in Environmental Sustainability* 7 (2014): 22-27.

National Research Council. *Improving Disaster Management – the role of IT in Mitigation, Preparedness, Response and recovery*. The National Academy Press: Washington, USA. 2007

Pfister Jr, Paul W. *Humans and Their Impact on Cyber Agility*. AIR FORCE RESEARCH LAB ROME NY INFORMATION DIRECTORATE, 2012.

G. Riley, M. Ammar, "Simulating large networks: how big is big enough", in *Proc. First Int. Conf. on Grand Challenges for Modeling and Simulation*, San Antonio, TX, 2002.

W. H. Sanders and J. F. Meyer, "A unified approach for specifying measures of performance, dependability, and performability," in *Dependable*

Computing for Critical Applications, Vol. 4 of Dependable Computing and Fault-Tolerant Systems, A. Avizienis, H. Kopetz, and J. Laprie, Eds. Springer-Verlag, 1991, pp. 215–237.

Schramski, J.R., Kazanci, C., Tollner, E.W., 2011. Network environ theory, simulation, and EcoNet 2.0. *Environ. Model. Softw.* 26, 419–428.

Shapiro, C., Varian, H. R. *Information Rules: A Strategic Guide to the Network Economy* (Boston, MA: Harvard Business School Press, 1999), 183.

Shen, Z., Gao, L., and Kwiat, K. 2003. “Modeling the Spread of Active Worms,” In *Proceedings of INFOCOM-2003*.

K. S. Trivedi, *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*, 2nd Edition. John Wiley and Sons, New York, 2001.

Whiteman, B. 2008, “Network Risk Assessment Tool (NRAT)”, *IA newsletter*, Vol 1, Spring 2008,

Yu, Y. (2013). *Resilience Strategies for Network Challenge Detection, Identification and Remediation*.