Cyber-Forces, Interactions, Terrain: An agent-based framework for simulating cyber team performance

Geoffrey B. Dobson

CMU-ISR-22-103
August 2022

Institute for Software Research
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Kathleen M. Carley (Chair)
Christian Lebiere
Greg Shannon
Leslie Blaha (Air Force Research Laboratory)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Societal Computing

This work was supported in part by the AFRL under Cyber-FIT grant (FA86502126244). Additional support was provided by the center for Computational Analysis of Social and Organizational Systems (CASOS) and the Center for Informed Democracy and Social Cybersecurity (IDeaS) at Carnegie Mellon University. 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 AFRL, or the U.S. government.

Keywords: agent-based modeling, simulation, military, cyber warfare, Cyber-FIT

Abstract

Over the past decade, militaries all over the world have declared cyberspace a domain of war. This has led to the professionalization of cyber teams in combat arms. These cyber teams must understand how they control, defend, and maneuver within this new domain, cyberspace. Moreover, commanders must train and equip the cyber teams to be successful with clear expectations of tasking and success criteria. The definition of success, for cyber team performance, continues to be a very evasive concept. Military and private organizations are spending a large amount of resources on training cyber teams. The training comes in many forms, ranging from individual skill development to advanced tooling, to team-based exercises. Due to the lack of understanding of cyber team performance measures, determining which training is most beneficial is nearly impossible.

This research seeks to bridge the gap by 1) computationally defining the performance measures of cyber teams and 2) creating a software tool that can simulate the deployment of cyber forces into conflict. This would allow researchers to experiment with a multitude of variables such as team makeup, training status, adversary type, organizational interaction, behavioral theory, and cyber terrain factors. To accomplish this, I created the Cyber-FIT agent-based simulation framework. I describe all versions of the software, built in spiral development methodology to arrive at an architecture that can realistically simulate a cyber team deployment. I present the definitions and real-world applicability of the performance measures of cyber teams. A realistically scaled cyber conflict is simulated in order to collect synthetic data and analyze all of the measures. Several virtual experiments are conducted using the model to show its usefulness, along with a sensitivity analysis of the most important control variables. Finally, a validation of the Cyber-FIT model is presented.

Acknowledgements

This educational adventure of a Ph.D. at Carnegie Mellon University has been one of the most rewarding experiences of my life. First and foremost I have to thank Dr. Kathleen Carley for taking me on, a part-time student, and allowing me the flexibility and pace that made it possible. Dr. Carley was so helpful to me over countless whiteboarding sessions, and working through which research questions were most applicable, and interesting. I also want to thank Dr. Jeff Boleng, former CTO of the Software Engineering Institute in his encouragement early on in the program. Thank you to Dr. Greg Shannon for the breakfast research chats at Ritter's Diner. Thank you to Dr. Christian Lebiere for helping me to understand ways to model theories of mind. Thank you to Dr. Leslie Blaha for aligning my work with research objectives of the Air Force Research Laboratory. My committee was wonderful and I'm blessed to have met them and worked with them over the years.

I am grateful to all of the students and colleagues I met along the way, especially my mentor Matt Benigni who showed me the ropes early on. Thank you to all of the others who helped with brainstorming sessions, coding tips, presentation feedback, and thoughtful conversations, especially David Beskow, Stephen Dipple, Sienna Watkins, Sumeet Kumar, Iain Cruickshank, Prasoon Patidar, Luke Osterritter, Shannon Gallagher, Dustin Updyke, Tom Podnar, Chad Hershberge, Bill Reed, and Sam Perl. Thank you to Kim at Potomac Station Coffeehouse in Dormont, where I wrote the majority of my thesis document over many cups.

Thank you to my family for all the support throughout my life. My mother Lee Ann's employment at a university afforded me the opportunity to attend college and study math and engineering. My father Bruce embedded a constant search for knowledge in me, most notably by watching every episode of The X-Files together all those years back. Thank you to my daughters Riley, Reagan, Reese, Rowan, Ruby and Remi, you are the inspiration for everything I do. Finally, and most importantly, thank you to my wife Jessica for the unwavering support, encouragement, and love.

Contents

1 Chapter 1: Introduction

1.1 Cyber is a domain of war

The starting point of this work came in 2011 when the United States Department of Defense (DoD) declared cyberspace a domain of war adding it to land, air, sea, and space [1]. The purpose of the military is to control domains, so cyber must be treated in a similar fashion to the others. This means understanding how to operate and maneuver within domains in order to successfully engage an enemy. Consider a land-based engagement: an army commanding general officer will know exactly what land terrain they have control of. Now, the commanding general officer must know how much cyber terrain they are controlling, and what will prevent them from maneuvering further in cyberspace. This introduces an incredible amount of complexity to the battlefront. Cyberspace is difficult to conceptualize because it can be thought of in many ways such as physical connections, logical dependencies, or virtual networks, to name a few. This is a well know problem, first envisioned in 1980, describing the open systems interconnection (OSI) model [2]. Today, there is no standard showing the "cyber battlefield" to a commander in order to make decisions on how to maneuver in this domain. This problems seeps into nearly every other aspect of military operations in the cyber domain. According to Joint Publication 3 – 12, cyberspace is "A global domain within the information environment consisting of the interdependent networks of information technology infrastructures and resident data, including the Internet, telecommunications networks, computer systems, and embedded processors and controllers" [3]. This means that U.S. military planners must now determine how they will survey, secure, and protect the cyber domain, just like a land mass, an area of the ocean, or air space. This evolution of cyberspace, as a domain of war, has spawned a huge growth in "cyber teams" over the past decade. These teams are military units whose primary mission is to operate within cyberspace.

The United States government has continually increased funding for cyber teams over the past decade as cyber terrain blends into more and more kinetic missions. The budget request by the Pentagon in 2022 called for $11.2B in cyber funding [4]. This comes a year after the Pentagon budget increased the number of cyber teams by 10% [5]. Clearly, this is a very complicated and expensive problem for the military to solve. The figure below displays the increase in total cybersecurity spending by the United States federal government from 2017 – 2021.

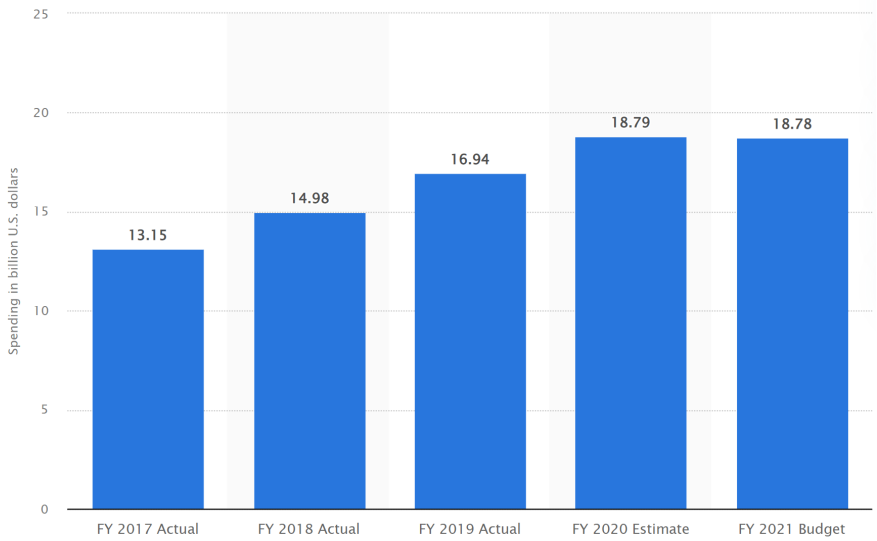


Figure 1: United States federal spending on cybersecurity 2017 – 2021 [6]

This rapid expansion into cyber is not limited to the military. Cyber crime is an enormous drain on society in the form of costs to private citizens and industry in general. The figure below shows projections of the cost of data breaches to the public alongside increased spending on cyber security products and services. In short, hundreds of billions are being spend on a trillion-dollar problem.

Figure 2: Cost of data breaches and cyber security spending 2017 – 2022 [7]

The DoD published its Cyber Strategy of 2015 calling for an increase of investments that can aid in understanding how cyber teams operate in the cyber domain. Specifically, it calls for the need to "establish an enterprise-wide cyber modeling and simulation capability", and more specifically, to "assess the capacity of the projected Cyber Mission Force to achieve its mission objectives when confronted with multiple contingencies" [8]. Clearly, this modeling and simulation capability would come in the form of software that must define cyber forces, terrain, and how the engagements would play out. This leads to the original ideations of how Cyber-FIT would work.

1.2 Modeling and simulation of cyber effects

Militaries have been war-gaming and analyzing the application of forces since the beginning of time. Modern war-gaming likely began in the early 1800s by Prussian commanders as a way to train young officers about the many unpredictable aspects of warfare [9]. The RAND corporation [10] studied papers and conference transcripts of the scientific war-gaming efforts in the 1950s and 1960s, attempting to project a thermonuclear war. The final analysis by the experts of the time was clear: simulation was the only way forward. Over the past several decades, increases in computing power have provided researchers the apparatus with which to fulfill the promises of simulation. In the 1970s, the field of agent-based systems emerged, most notably with Conway's game of life [11]. Agent-based modelling is a technique, or subset of simulation, where entities are defined as agents, which have behaviors dictated by rulesets. Agent-based modelling is especially useful for studying complexity such as emergent behavior from social systems [12]. Bonabeau [13] describes four categories of simulation most suited for agent-based modelling: flow simulation, organizational simulation, market simulation, and diffusion simulation. Organization simulation can be of many types such as makeup of personnel, personnel number, organizational constraints, communications structures and systems being utilized. This thesis will be a deep exploration of organizational simulation as applied to cyber teams.

Due to the nature of most military operations, where units are working towards a goal, made of autonomous agents, agent-based modelling is a natural fit. The unit can be seen as a complex system, with varying levels of similarity, involvement, perception, and performance amongst individuals. A survey [14] of agent-based modeling literature from 1998 – 2008 showed that 13.6% of peer reviewed papers were on military applications. Efforts have increased at improving cyber simulation models. An early information warfare simulation framework [15] was proposed and prototyped by Welsh, Conti, and Marin that connected objects of interest through partially ordered discrete events. The framework laid out an effective concept of interactions amongst nodes that hold and pass information but does not account for the human-to-computer, and human-to-human interaction necessary to conduct behavioral and social simulation. Bergin [16] proposed a

cyber-attack and defense simulator that accurately models all aspects of wireless network and autonomous vehicle behavior. The framework data definitions include proper OSI layer attributes, packet-based simulations, and configurable control variables. Simulation software with this level of fidelity will be very expensive to build and maintain, especially as the technology used in the field will continue to evolve. Therefore, design considerations should be made that prioritize the most pressing issues for military organizations. Thompson and Morris-King created [17] a simulation framework specific to mobile tactical units which addresses the interplay between hierarchical command and control structure, group mobility, and cyber security. The model is quite useful for the study of mobile tactical units but doesn't generalize to other types of missions and forces well. One of the most useful aspects of agent-based modeling and simulation is conducting virtual experiments to compare various courses of action. Virtual experiments to analyze courses of action are especially useful when the alternatives are costly to implement. Simulating training strategies fits into this category, since the military spends such a large portion of its budget on educating, training, and exercising its forces. Petty, Barbosa, and Hutt implemented [18] an agent-based model to simulate the cost of three different live, virtual, and constructive training approaches. They estimated labor costs for all personnel involved in building and delivering each training system and determined which alternative would be least costly to the Air Force.

While there are many agent-based models for military applications, none exist that are specifically designed to simulate the behavior, operations, and ultimately, the performance of cyber teams. In this thesis, I will create an agent-based modeling and simulation framework that addresses the gap in understanding about how cyber teams perform in missions. The software will model cyber team behavior and project team operational outcomes that the DoD is calling for.

1.3 How cyber teams are trained, equipped, and assessed

Cyber teams, being a relatively new construct within the United States military, have been an ongoing challenge to train, equip, and assess. Training is a military necessity which is outlined in a multitude of joint and service specific doctrine. The Chairman of Joint Chiefs of Staff Instruction 3401.02B states "Units will report the present level of training of assigned personnel as compared to the standards for a fully trained unit as defined by joint directives" [19]. This training level is considered a "T" rating. A fully trained unit typically has a level of knowledge and experience required for each personnel billet. This language is broadly applicable to all United States military units. So, cyber units all have a training rating that would associate their relative level against an ideally trained team. So, then, what would constitute an ideally trained cyber team? This would likely be a combination of basic cyber knowledge with job specific training. Military personnel each have what is called a military occupational specialty (MOS) which

prescribes job specific training for those duties. Cyber specialty training in the form of information security, coding, architecture, etc., is highly attractive for new recruits and is likely a reason that cyber soldiers have better retention throughout their initial contract terms [20]. Cyber personnel, after completing their MOS technical training frequently are able to receive follow on training through private providers such as CISCO and SANS institute. Therefore, a list of individual trainings associated with a particular billet, aggregated to a team level would be that ideally trained unit. This would then be coupled with the requisite experience level and ranks that fill out the typically hierarchical military organization. The teams were first rolled out in 2011 by United States Cyber Command with a squad hierarchy and mixture of ranks reporting to a headquarters element. Each "cyber protection team" was made up of thirty-nine personnel billets broken up into five squads: mission protection, discovery and counter-infiltration, cyber threat emulation, cyber readiness, and cyber support [21]. This served the purpose of breaking specialties into different roles and associating specific training with different squads and individuals. Along with individual training, cyber teams, like all U.S. military units, must complete periodic team-based exercises, inspections, and evaluations. These collective training events ensure that the individuals within the unit can combine tactics techniques and procedures, in concert, to meet unit level mission essential tasks.

Equipping cyber teams has also been very difficult. A recent Air Force report found a multitude of problems such as poor contractor support, sub-optimal cyber training ranges and non-standardized tooling [22]. A top-down approach is likely not appropriate due to the decentralized nature of cyber teams along with differences in missions. Simply searching for cyber security tools provides an overwhelming number of options. This leads to cyber teams using various open-source tools, their own custom code, and potentially dangerous non-sanctioned executables from the web. It's very hard for official government offices to track all of the options much less vouch for a recommended toolkit. Big technology companies are vying for large contracts in this space and competing with each other to provide such capabilities, most recently in secure cloud systems [23]. Software simulation can help address this problem by defining what cyber teams are doing, and how those tools assist in meeting the mission essential tasks of units

Of all the commander's tasks, assessing cyber teams is probably the most difficult job at this time. Unlike other domains of war, where it's visually apparent where a unit is physically, and how much damage has been done to equipment and capabilities, cyberspace is mostly invisible. Cyber commanders must rely on the reports of subordinate units and dashboards that are reading bits from networked computing systems. The United States military is commander centric. A large amount of responsibility is placed on commanders to use their judgement to assess situations and apply commander intent. This concept applies to the assessment of forces. This is clearly stated in the Commander's Handbook for Assessment Planning and Execution: "Assessment is a key component of the

commander's decision cycle, helping to determine the results of tactical actions in the context of overall mission objectives and providing potential recommendations for the refinement of future plans" [24]. The figure below shows a visual representation of the basic steps and flow of the continuous nature of assessment to support commander decision making.

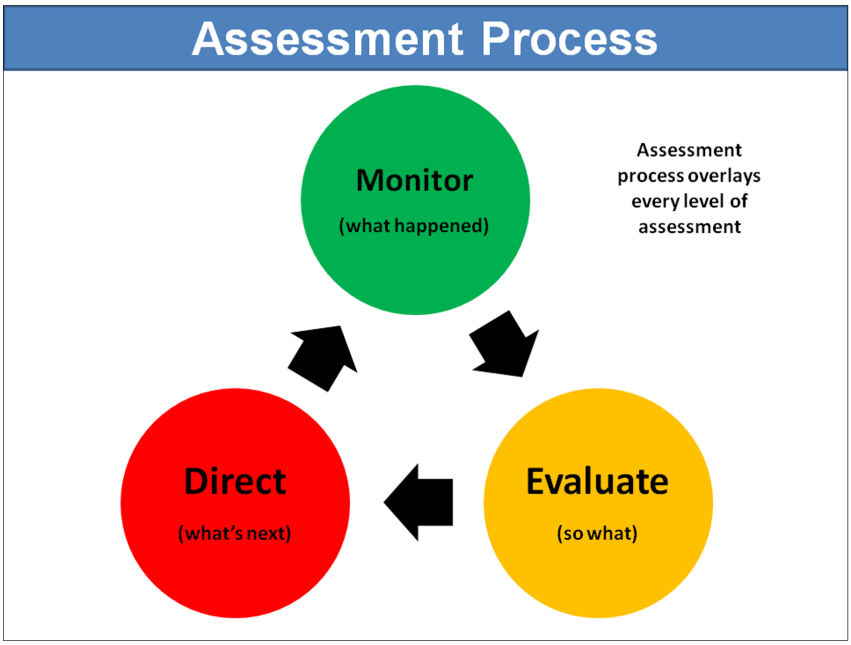


Figure 3: Assessment process overview as defined by Joint Staff J-7 [24]

The assessment of training is also doctrinally defined by the Joint Staff as part of the four-phase assessment of the joint training system methodology. According to the Joint Training Manual of the Armed Forces of the United States, the purpose of assessment is "to determine which organizations within the command are able to perform at the level required to meet the task standard(s), and which missions the command is trained to accomplish" [25]. Higher level commands break down missions into smaller pieces that are accomplished by subordinate units. Frequently, collective training events in the form of cyber war exercises are used to determine which missions the commanders are proficient in.

1.4 Gaps in the state of the art of cyber training and assessment

Ultimately, cyber leadership throughout the military is charged with training cyber teams, and then assessing those training efforts. There are clear gaps in the state of the art of training and assessing cyber forces. Consider the figure above detailing the continuous process of assessment that commanders must undergo. Take an illustrative example where a commander will monitor an operation, then evaluate the performance of the team, and

then direct activities as a result. Perhaps this operation is completing cyber related information requests (a typical task within a cyber audit, inspection, or survey). The figure below applies this cyber operation to the doctrinal assessment flow and details the required data that must be known to the cyber commander to effectively move through the assessment flow.

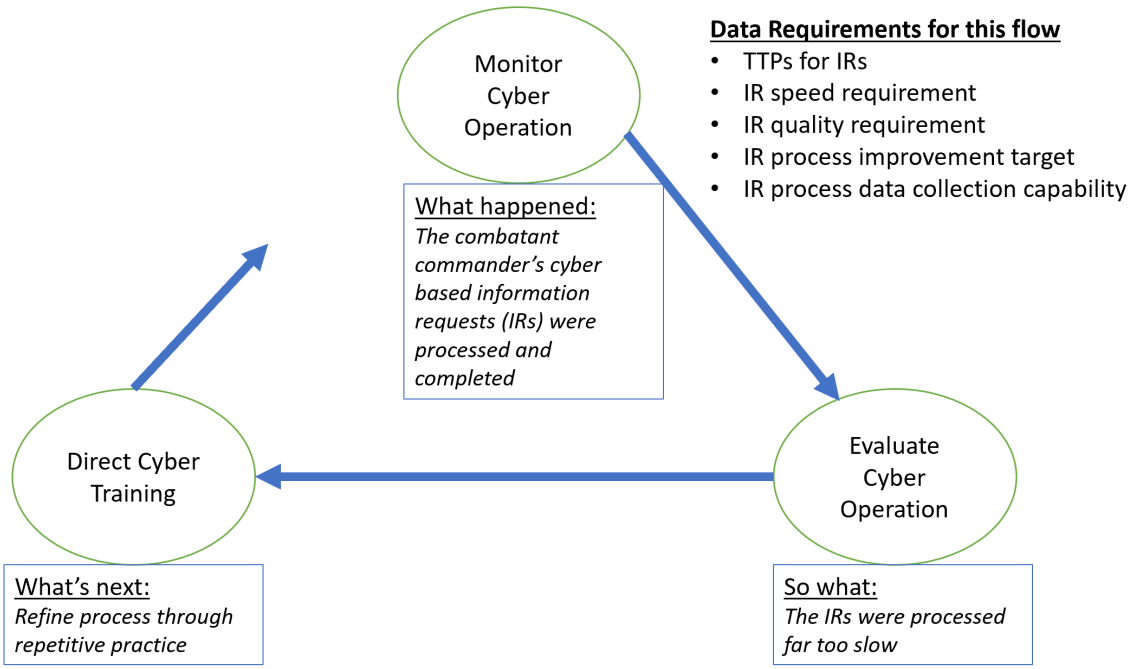


Figure 4: Data requirements for illustrative assessment flow

As the commander monitors the operation, the required tactics, techniques, and procedures (TTPs) associated with this operation must be clearly understood. This would be all of the processes mapped out along with the required interactions amongst squads and team members where information is exchanged. The type of information being exchanged would be known, along with the characteristics of the data and the medium of exchange. Now that the operation is complete and many information requests were processed and fulfilled, evaluation data could be considered. In order to properly evaluate the operation, the commander would need to know what the qualitative and quantitative requirements were for the information requests. Ideally, data would be collected about the TTPs utilized by the cyber team for post-analysis. With evaluation data on hand, and a proper evaluation completed, the commander could then direct cyber training targeted at the parts of the operation that did not meet standard which would include improvement targets in terms of quality and quantity (performance measures). This same process could be applied to any number of cyber operations such as defending key- terrain cyber, implementing new security controls on a network, or emulating a cyber adversary.

Taking one more illustrative case example, this same process could be applied to the commander's assessment of a cyber war exercise which is widely regarded as the best way to put a cyber team to the test. The most critical TTPs would be mapped out with clear metrics delineating successful defense of cyber terrain. This would include details about how the cyber team must harden the protected terrain, hunt for adversary presence, and remediate threats and active attacks, which would all be monitored during the exercise. All of this data, during the exercise at prescribed intervals, and then at the end of the exercise would be evaluated against standards that include data and definitions of the expected performance of those TTPs. Finally, the commander, with advise from senior technical advisors, would conduct a thorough after action review (AAR) where TTPs would be examined in earnest. TTPs might be redeveloped, tweaked, or tossed away. The commander could then direct new training that would focus the team's energy on the most glaring problems. The table below summarizes some of the most essential data and definitions that would be required for the cyber commander to work through the assessment flow in the second illustrative case about a cyber war exercise.

Step Name	Data and Definitions Needed for Performance Measures
Monitor	How should the team react to a found advanced persistent threat? Who should be notified, with what information? Which personnel should be securing the network? Which personnel should be hunting for adversaries in key-terrain cyber? How much of the network needs hardened? What vulnerability level is acceptable for this mission? What mission data determines if they team should move to the protect phase?
Evaluate	How quickly should the team find the adversary within the network? How much evidence is acceptable to claim a particular host is compromised? How quickly should a team report to higher headquarters once an adversary is detected? How many missed vulnerabilities is acceptable during the survey phase of the mission? How quickly should a team be able to restore compromised cyber terrain?
Direct	What is the average cyber team time to react?

	What is the average cyber team time to restore? What is this team's typical performance metrics? Which TTPs were changed for this exercise? Which relevant training activities affected the team's performance? How were leaders exercised in tasking and communications for this exercise?

Table 1: Questions to elicit data and definition requirements for cyber assessment flow

These two illustrative cases are meant to highlight the key gap in the state of the art for cyber operations that this thesis is addressing: the data and definitions for performance measures. Put simply, commanders do not know how well their cyber teams are performing. The reason is that the data and definitions which would indicate performance are not known. For example, in 2018, the Army requested [26] $429 million in funding for a new cyber training range capability. The cyber range will have capabilities to simulate "real-world mission rehearsal" and necessary friendly, supporting, and adversarial forces. Many of these concepts are quite clear to the developers and personnel building the cyber range systems. What is not clear, is how cyber mission forces are assessed once they enter into the range for scenario engagements. On September 26, 2018, at the Joint hearing to receive testimony on the cyber operational readiness of the Department of Defense [27], Brigadier General Dennis Crall, principal deputy cyber advisor at the Office of the Secretary of Defense said the following: "I would say we need to ensure that we have a solid baseline and assessment mechanism so, when we come back here and talk to you about what's working and what's not working and how we've spent money, we can do so with the right kind of accountability". That is, currently, there is no baselines mechanism to know how well cyber forces are prepared for their missions. For individual personnel, we do have some semblance of their experience, knowledge, and skills, by simply knowing what education and certifications they've completed and how many years experience they have. When we aggregate those skills and experience into teams, it is very difficult to compare teams, and predict how well given teams would do in given mission sets. Furthermore, there is no team-based assessment baseline, to understand objectively, who the elite cyber forces are.

This gap was also called out in the Defense Science Board Report of 2013 called "Resilient Military Systems and the Advanced Cyber Threat" [28]. This report was the summary of a task force of senior security analysts and scientists' findings on the state of cyber security in the military as it would be able to compete with peer adversaries. The report states "The Task Force unsuccessfully searched for cyber metrics in commercial, academic and government spaces that directly determine or predict the cyber security or resilience of a given system". This point underpins the ongoing and yet still unsolved

systemic problem in the cyber security industry: it is extremely difficult to grasp true cyber situational awareness of a given enterprise, system, team, or set of circumstances. Furthermore, the report states the "Department will do best to measure outcomes, such as the average time it takes to detect a successful attack that breaches the network perimeter defenses, and the amount of time it takes to recover a system this is lost as a result of a cyber attack". These types of measurements are precisely what is needed to guide performance metric development.

Ideally, performance measures are proposed, computationally modeled, simulated, empirically observed, assessed against the simulated data, and then refined in a continuous iterative loop. In fact, this is the only way it can be done. The task force created a notional dashboard of cyber team performance metrics that would be aggregated at a headquarters type organization which is shown in the figure below. A primary goal of this thesis is to simulate a similar dashboard. By simulating this dashboard, the processes, data structures, and cyber team performance formulas would be defined and computationally modeled.

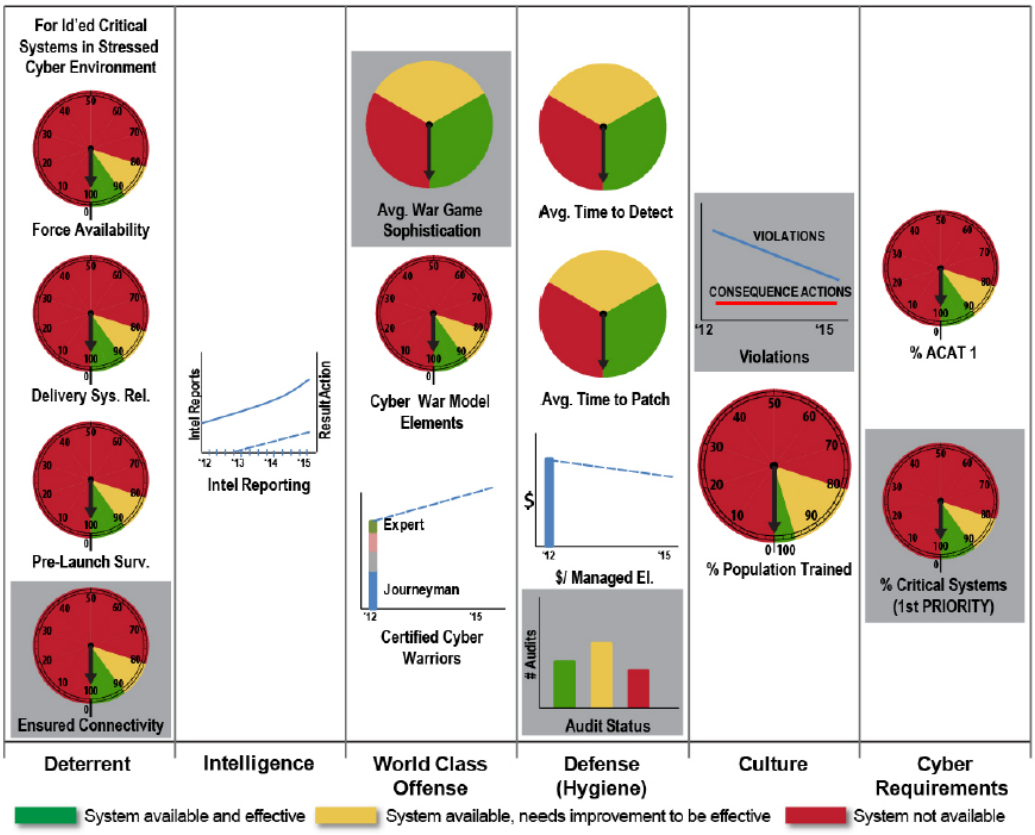


Figure 5: DSB notional cyber team performance dashboard

One last important United States government document that identified this gap was published in 2019 which is the White House Executive Order on America's Cybersecurity Workforce [29]. This document outlined many objectives for the administration in the

form of resource allocation towards strengthening the cyber skills of the American government. The order states: "The Secretary of Homeland Security, in consultation with the Secretary of Defense, the Director of the Office of Science and Technology Policy, the Director of OMB, and the heads of other appropriate agencies, shall develop a plan for an annual cybersecurity competition (President's Cup Cybersecurity Competition) for Federal civilian and military employees. The goal of the competition shall be to identify, challenge, and reward the United States Government's best cybersecurity practitioners and teams across offensive and defensive cybersecurity disciplines". For the competition to be successful, organizers must create a scoring system that would reward actions taken to mitigate cyber threats. The order doesn't explicitly call out the point system, or what guidance it should follow in the form of national level cyber frameworks such as those published by the National Institute for Standards and Technology. This omission seems to show that the metrics are still unknown at the time of the order.

1.5 Other gaps associated with cyber team performance

The primary purpose of this thesis is to create a software tool that simulates a cyber team engagement, so that performance measures can be computationally modeled. The previous section described ways this gap is described in various United States government documents. In order to build a software that addresses the primary gap, other secondary gaps quickly emerge that have to do with the inputs, behaviors, and outputs of such a software simulation tool.

For instance, consider input data that the software would rely on. It would likely seed the simulator with data to describe the environment that the cyber team would fall into. Also, basic demographic data about the cyber team would be included such as skills, certifications, rank, position, experience, education, and military occupational specialty, akin to a readiness roster. The simulator would have to ingest adversary data that differentiated complexity and behaviors, which is similar to an intelligence threat briefing. If the cyber team is deployed to support kinetic missions, then friendly forces data would be included about the missions, personnel, user profiles, and associated hardware and software requirements. Cyber policy data might be ingested that details information differentiating communication requirements, access details, cyber terrain fielding guidelines, maintenance schedules, and cost limitations. Behaviors built into the model would be driven by data as well. Most agent-based behaviors within a model are based on stochastic variables which infuse randomization into the model. There are a plethora of candidate behaviors to be modeled into a cyber conflict simulator. Those most important for the purpose of generating performance measures are: cyber operations, friendly force terrain usage, human system, and computer network behavioral modeling.

Finally, when considering the output data that would be generated as a result of this model there are other tangential data of interest that should be based on real world

operations as well. The modeling environment itself would ideally be able to export data in a controlled manner where the performance measures can easily be computed and stored.

Input Data	
Item	Description
Base Cyber Terrain	Network architecture diagram listing computer systems that support the base infrastructure. This includes networking devices, servers, hosts and associated hardware and software descriptions
Kinetic Missions Supported	Summary data of kinetic missions that the base cyber infrastructure is supporting in terms of friendly force missions. This data includes number of forces and cyber terrain that the missions depend on and are actively utilizing
Cyber Team Rosters	Demographic and positional information about the cyber team including rank, MOS, squad, certifications, education, and experience
Adversary Intelligence	Information pulled from a data source similar to an intelligence threat report with adversary complexity, indicators of compromise, code names, recent activities, and hash values
Cyber Policy Data	Information pulled from cyber policy statements that are in effect and applicable to the simulation such as segmentation, building physical security, training, encryption, communication security, operational security, industrial security, classification levels, data management, and personnel security. This can also include environmental data such as location, connection to private sector, internet service provider, force protection conditions, and expeditionary details
Behavior Data	
Item	Description
Defender Cyber Operational Behavior	Model of basic cyber operational behaviors that a defender would undertake such as surveying terrain, collecting vulnerability data, removing vulnerabilities from systems, communicating with team members, and restoring compromised cyber terrain. Behavior should be tied to military occupational specialties such as communications security, threat emulation, network transport, host security, forensics, development, architecture

Attacker Cyber Operational Behavior	Model of basic cyber operational behavior that an attacker would undertake such as moving through the cyber kill chain of reconnaissance, resource development, payload delivery, compromise, command and control, and actions on objectives. Realistic simulation of how the attacks will work at a low level similar to details found in the MITRE ATT&CK database
Friendly Force Cyber Operational Behavior	Model of cyber operational behavior expected from friendly forces utilizing cyber terrain
Human Behavioral Modeling	Various selected human behavior models that can inject stress, tiredness, patterns of life, teamwork, etc.
Cognitive Modeling	Various selected cognitive models that can inject mistakes, forgetting, awareness, transactive memory, etc.
Cyber Terrain Network Behavior	Network diagrams of the cyber terrain including information such as routing, virtual local area networks, intrusion prevention systems, subnets, servers, hosts, and special enclaves. Models of how different cyber terrain react to stimulus from the operational environment, store information, and interactions with both human agents and other terrain agents
Output Data	
Item	Description
Performance Measures	Formulas for computing performance measures of cyber teams. This could include agent-based data on defenders, attackers, and friendly forces. Cyber terrain data will be of significant interest as it relates to how the systems are operating in terms of access, security, vulnerabilities and if they are compromised. Finally, network-based data should be output in terms of interactions between nodes and links
Modeling Environment	Integrated development environment that allows researchers to experiment with changes to behaviors, data structures, input configurations, and environment variables to extend existing cyber team simulation capabilities
Data Collection	Data collectors that bin and categorize data appropriately along

and Processing	with processing software to store performance measures and output data upon completion of runs of the simulation. Processing would also include concepts that are important to simulating conflicts and wargaming such as mission battle assessments and after action reviews.

Table 2: Summary of ideal input data, behaviors, and output data for a robust model of cyber engagement

Viewing all of the different data and modeling in the table above that is necessary for a robust cyber simulation tool shows how vast the secondary gaps in the state of the art and science are. While examples of what is listed can be found in literature, the fact that the Department of Defense is still requesting software that aggregates and models the system as a whole show that most are one-off projects that don't integrate with other existing simulation software. Ideally, a realistic and robust cyber effects modeling and simulation tool would include everything listed in Table 2 and more. This work will include as much as possible within the scope necessary to accomplish the primary goal of a realistically scaled simulation of a cyber team with all performance measures output.

The software research work in this thesis will provide a way to computationally compare teams and run virtual experiments that allow the user to ask what-if questions on how to approach cyber team construction. A capability like this would help provide the accountability that lawmakers are requesting which is to validate the expenditure of such large resources. The primary contribution of this work is in the metrics and methodology to computationally model cyber team performance. Brigadier General Crall stated the need for a "baseline and assessment mechanism". That can only be done by defining the metrics and measures of cyber teams, and then laying out a systematic and repeatable mechanism with which to compute them.

1.6 Goals of this thesis

There is clearly a gap in the understanding of how cyber teams perform in missions and what their contribution to military goals are. This thesis work improves our understanding by providing a computational and quantitative projection of what the cyber forces are doing. To accomplish this goal, three high-level milestones must be met: 1) define cyber team performance measures, 2) create an agent-based software framework to simulate performance outcomes, and 3) validate the software. Chapter two will describe the iterative early work where the model was built from scratch and how the early versions helped understand how to define cyber team performance measures. After more work interacting with cyber teams and discussing outcomes with subject matter experts, chapter three describes the computations and formulas that define cyber team performance. Chapter four describes the current version of Cyber-FIT which is able to simulate conflict

and compute all of the performance measures. This chapter also describes a realistically scaled simulation, model sensitivity analysis and two virtual experiments. Chapter five walks through the most common agent-based model validation methods and applies those techniques to Cyber-FIT. Chapter six describes the how far this work went, limitations, and future direction.

There are two immediately useful generalized use cases for this software: wargaming and virtual experimentation. (Both of these use cases are military focused, but like many other concepts throughout this work, it can easily be applied to industry by adjusting the outcomes and focus of simulations.) The first use case of wargaming is at the strategic level. Higher ranking military officers are responsible for campaign planning where large number of forces are deployed to accomplish specific military objectives. A wargame will almost always be multi-domain and joint. Multi-domain means several or all of the domains of war (land, air, sea, space, cyber) are in play, and joint means several or all of the service components (Army, Air Force, Navy, Marines, Space Force, Coast Guard) are utilized. A wargame moves through turns where scenarios are presented, options are selected, and then a simulation presents the results and new challenges that must be addressed. Consider a situation where the participant selects to move an air component package, on a carrier, into a contested area of the sea. The turn might simulate how the adversary responded by attacking industrial energy capacity. Now the participant has to respond to that new scenario. Wargaming is very difficult and resource intensive. Cyber-FIT could be used in a wargame simulation by providing the simulations of the cyber domain outcomes that a participant chose. For example, the participant might be forced to choose whether to deploy a cyber team immediately, or hold off for a turn, to engage an unknown enemy in contested cyber terrain. This exact use case is the subject of virtual experiment two in chapter four of this thesis.

The second use case of virtual experimentation is at the tactical level of cyber conflict. This is the area where unit level cyber leaders are making difficult decisions with incomplete information. Cyber leaders at this level are dealing with issues such as training options, deployment planning, readiness preparations, and squad assignments. All of these decisions ultimately result in how well the team performs when deployed to a conflict. When talking with cyber leaders, at the tactical level, many have expressed interest in a software that would simulate some (or all) of these decisions to help think through the tradeoffs of interest.

At the end of this work, a cyber mission planner could look at a schedule of upcoming missions and set up a virtual experiment where the teams being assigned to missions would be simulated and an assessment of the plan could be carried out. Currently, no projections like this are being done through a software simulation. Virtual experiments of this nature will be able to be conducted with Cyber-FIT at the end of this thesis. For example, a team leader might be observing an average cyber team within the organization.

This team skill level could be improved by adding an expert skill level troop or providing enriching exercises that takes one of the existing team members from average to expert skill level. The former path means that the expert is taken away from another team within the organization. The latter path means that the team must wait until the skill has been acquired which takes both time and money. There are tradeoffs for both choices and the other choice is to keep that team status quo. This use case is the subject of virtual experiment one in chapter four.

Simulating cyber effects in a realistic manner and meaningful way is extremely difficult. If this is a critical need for the United States military, why hasn't it been done? I think the answer to that question is the fact that it is so difficult. Government contracts are usually awarded based on requirements. The requirements are so difficult to write that work like this is hard to specify. If a government agency wanted to acquire a software that projected cyber mission effectiveness, the contractor might first say: Please define effectiveness. There is a chicken or egg effect occurring in this realm. The data to support simulations is needed. But the simulation software needs data to determine if it's simulating the right thing. The barrier of entry also may be return on investment. It would be hard to justify an investment by a private contractor for a simulation software where the outcome is not clear. There is a significant effort required to formulate measures, develop a computational model, and analyze results, all without a clear pay off. Luckily, this is the perfect subject for a doctoral thesis.

2 Chapter 2: Cyber-FIT versions 1 – 3

The Cyber Forces, Interactions, and Terrain agent-based simulation framework began as an attempt to understand what the most basic actions of cyber teams are. Terms like defensive cyber operations, and network hardening are used frequently when describing cyber team activities, but don't define the precise connections amongst team members and the computer networks at large that would describe these actions. Cyber-FIT will define this phenomenon using agent-based technique in the form of agents, rulesets, and interactions. In this section, these concepts will be introduced at the ground level and then built up as more complexity as added.

Modeling cyber warfare has proven to be very difficult. There are a multitude of variables, many of which are either dependent on the specific situation encountered, or difficult to measure. Furthermore, in most cyberspace environments, where it is already difficult to quantify known entities, there are unknown entities that may affect the behavior of the systems. An agent-based modeling and simulation approach will be taken throughout the entire thesis to investigate the behavior of cyber teams and extend existing computational organizational theory. An agent-based modeling approach is being applied because, as McCall and North [30] describe: "the systems that we need to analyze and model are becoming more complex in terms of their interdependencies". Clearly, the operations of cyber teams, in cyberspace, are complex. Personnel, environments, technology, missions, team size, policy, adversaries, and tools, just to name a few, can all be represented, in a near limitless way. Therefore, careful considerations must be made in regard to how different parts of the complex system will be designed. As Bonabeau [13] points out, "a general-purpose model cannot work. The model has to be built at the right level of description, with just the right amount of detail to serve its purpose; this remains an art more than a science". The art of the Cyber-FIT design and development will be addressed throughout, as simplistic behaviors become more complex. Typically, the minimal effective dose of complexity will be sought, addressing a new research question, in a spiral development methodology. A visual representation of the framework is provided in the following figure.

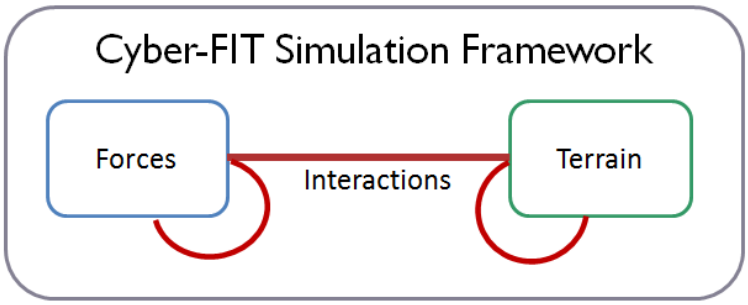


Figure 6. Visual representation of Cyber-FIT framework

The agents can be encapsulated, allowing for rulesets that define their behavior within the simulation world. The rulesets depend on, and respond to agent variables, environment variables, and interactions that occur. By assigning characteristics to the forces, interactions, and terrain, outcome variables of simulated cyber engagements can be projected. Also, the characteristics of the components of the simulations can be altered in order to differentiate various courses of action. The original goal of the framework is to take a first step toward providing the modeling and simulation capabilities requested in the DoD Cyber Strategy [31], Defense Science Board Report [32], and White House Executive Order [29]. Overall, it is a holistic approach to conducting experiments about the interaction of cyber terrain and forces.

There are two main classes of agents: force and terrain. Force agents are the military and non-military personnel engaged in conflict. Terrain agents are all of the computing systems being utilized for that conflict. The following table lists the agent types and sub-types of Cyber-FIT.

Agent Type	**Sub-Types**
Force	Defensive, Attacker, Friendly
Terrain	Networking, Server, Client

Table 3: Agent Types of Cyber-FIT

Agents interact with each other, which are directed links. There are three types of directed link interactions defined in Cyber-FIT: force-force interactions, force-terrain interactions, and terrain-terrain interactions.

2.1 Cyber-FIT version 1

Cyber-FIT version 1 was designed and developed using the NetLogo agent-based simulation scripting tool maintained by Northwestern University [33]. Version 1 was designed to get a model working where the earliest research questions around quantifying

cyber conflict outcomes could be addressed. This includes over-time (temporal) variables. As a cyber conflict goes on, it is necessary to know how much more vulnerable terrain is becoming and then how much it has been damaged. Similarly, military leaders want to know if the cyber forces are effective in gaining cyber terrain integrity. This leads to the first decision points in design around where to deploy terrain, and what different forces are doing when interacting with that terrain.

2.1.1 Terrain

Terrain is defined as the computer systems that military units depend on to execute their assigned mission. This version of Cyber-FIT delineates three terrain types, as defined in the following table.

Terrain Type	Name	Summary Description
1	Networking	Networking systems such as routers and switches
2	Servers	Server systems such as web servers, domain controllers, file servers, and intrusion prevention systems
3	Hosts	User systems such as personal computers, devices, and tablets

Table 4: Cyber terrain types

The different terrain types will become vulnerable at different rates. The vulnerability rates were computed by taking the known number of vulnerabilities on each of the terrain types from a sample of systems from MITRE's common vulnerability and exposures database, an industry standard for defining, assigning, and tracking vulnerabilities [34]. The vulnerability rates are associated with a probability based on the relative number of known vulnerabilities, as described in the following table.

Terrain Type	Percentage of sampled vulnerabilities
1	14%
2	28%
3	58%

Table 5: Terrain vulnerability sampling

The different terrain vulnerability rates will also be affected by the environment that they are deployed in. This distinction between types and what rate of vulnerabilities

is not meant to be an exact match with real world. Instead, this is a reasonable approximation in order to create differential behavior which will then present emergent behavior to consider. The version 1 model defines three environment types that represent common military areas of responsibility. The environments are "base", "tactical", and "industrial". The table below provides a description of the three environments.

Environment	Summary Description
Base	The Base environment refers to a long-term fixed military installation
Tactical	The Tactical environment refers to a temporary military installation stood up for the purpose of an overseas conflict
Industrial	The Industrial environment refers to a non-military facility that controls an energy production operation the military depends on

Table 6: Terrain environment descriptions

The different environments will affect how quickly systems become vulnerable, by terrain type. Based on discussions with vulnerability experts, the terrain types were scored relative to each other, to determine within which environment vulnerabilities seem to appear at higher or lower rates. The table below defines the relative vulnerability rate across the three environments and details the probability that the system in that given environment will become vulnerable at any time. This information is incorporated into the software that determines if a given terrain is vulnerable at any given time. That is, in a cell labeled "High", the probability of a system moving from non-vulnerable, to vulnerable, is equal to the relative share of common vulnerabilities and exposures (CVEs) as previously described. In a cell labeled "Medium", the probability is reduced 50%. In a cell labeled "Low", the probability is reduced 50% again.

Terrain Type	Base	Tactical	Industrial
1 (Networking)	Low	Medium	High
2 (Servers)	Low	High	Medium
3 (Hosts)	High	Medium	Low

Table 7: Environmental effects on vulnerability growth rate by terrain type

2.1.2 Forces

Forces are defined as the military members that are deployed to the military scenario. The defensive forces are deployed with the purpose of protecting the assigned

cyber terrain. The user interface allows the operator to add any number of defensive forces, up to sixteen. The defensive forces will attempt to remove vulnerabilities that exist on the terrain at any given hour (each time tick in NetLogo). The defensive forces select vulnerable systems randomly, according to a schedule. At all hours, the forces defend Terrain Type 3, every third hour they defend Terrain Type 2, and every sixth hour they defend Terrain Type 1. This models the real-world constraint that servers and networking equipment can only be defended at certain times, e.g., when they are being patched. The offensive forces will attack the systems based on what type of attack is being launched. Version 1 supports three attack types that offensive forces can launch, as defined in the following table.

Attack	**Target Terrain**
Random	All Types
Routing Protocol Attack	Type 1 (Networking Systems)
Denial of Service	Type 2 (Server Systems)
Phishing	Type 3 (Host Systems)

Table 8 Attack types of Cyber-FIT version 1

2.1.3 Interactions

Cyber-FIT version 1 defines interactions as any instance when a force is actively accessing cyber terrain. In the real world this could be performing operations and maintenance, coding malware, applying patches, etc. Two types of interactions are modeled: offensive actions and defensive actions, which are limited to offensive and defensive forces, respectively. The defensive force agents will perform operations and maintenance activities, and then apply patches at every hour to a randomly selected vulnerable system. That system will become non-vulnerable following this interaction. The offensive force agents will attack randomly selected systems of the type associated with the attack selected, at every simulated hour. In order for a system to become compromised, it must be vulnerable at the time that it was attacked (an offensive interaction by offensive force agent). If vulnerable, then the system has a chance of becoming compromised (based on the exploitation success rate control variable).

2.1.4 Virtual experiments

Three virtual experiments were run using the first version of the model. Each experiment seeks to answer a specific question a military planner might have when planning cyber warfare operations. For each experiment, the motivation, and results are discussed.

2.1.4.1 How many forces should we deploy to minimize the effect of a routing protocol attack (RPA) in an industrial environment?

In this experiment, we are considering a specific attack (RPA), in a specific environment (base). We'll vary the number of forces from one through fifteen and examine the decrease on Type 1 system (networking) compromise rate. We're specifically searching for the number of forces, where, when adding one more troop, the projected compromise rate is within one standard deviation of the current projected force package effectiveness. We expect that as the number of forces increases, decrease in compromise rate will level off. Results are shown in the figures below.

As shown in Figure 7, we can expect a substantial increase in effectiveness moving from one troop to five. After five troops, the projected performance improvement tapers off. We still see improvements on the projected compromise rate of Terrain Type 1, our primary concern in this simulated mission, but it will be decreasing as we continue to add forces. To find the point when adding troops will make no difference at all, we search for the point where the increase in effectiveness is within one standard deviation of the current projected average Type 1 compromise rate. This is laid out in the table below. This point is found at forces = 11. At that point, the projected compromise rate is 4.64 with a standard deviation of 0.77. The projected compromise rate, when adding one more troop to the mission, is 4.06, within one standard deviation of the previous projection.

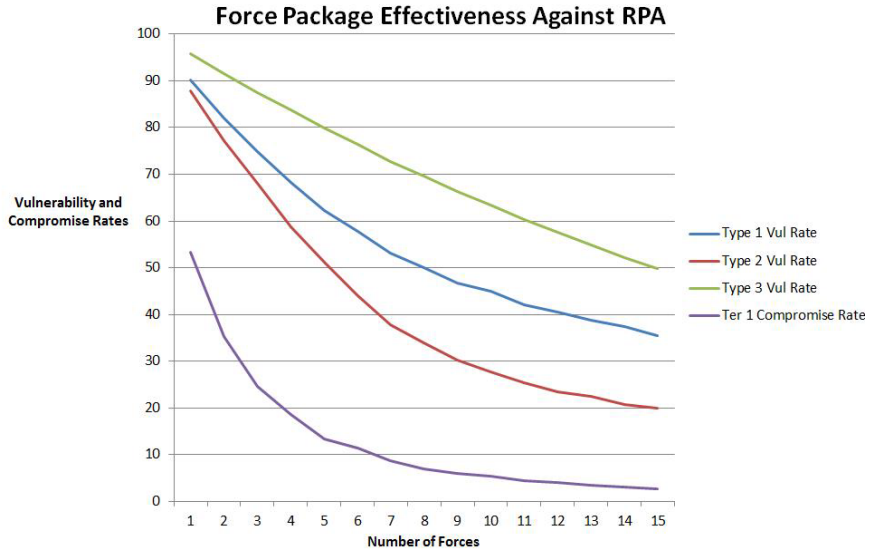


Figure 7: Virtual experiment results showing force package effectiveness

Forces	Compromise Rate	Standard Deviation
1	53.51	2.68

2	35.33	3.20
3	24.68	2.44
4	18.60	1.79
5	13.74	1.88
6	11.34	0.96
7	8.70	0.99
8	6.96	0.74
9	6.06	0.52
10	5.36	0.72
11	4.37	0.77
12	4.06	0.82
13	3.39	0.37
14	3.13	0.54
15	2.73	0.37

Table 9: Results of virtual experiment

This shows the importance of weighing the cost of adding more resources with the effectiveness of those resources. In this scenario, what do these numbers represent? We have a simulated mission on terrain that includes 21 type 1 systems. So, if the average compromise rate, at forces = 5, is 13.74, then we can expect, on average, 2.89 systems are compromised, when facing a routing protocol attack. At forces = 6, we can expect, on average, 2.38 systems are compromised when facing a routing protocol attack. So, somewhere between two and three systems will go down. Perhaps this is an acceptable risk? Also, once the attack is recognized, will five forces be enough to make an emergency change, repair the compromised terrain, and block the attack? This might be the case, which means that the planner should actually choose to deploy five forces, rather than eleven, due to acceptable level of risk, external constraints, and knowledge of mission resources.

2.1.4.2 What will be the expected effect on cyber terrain if the adversary switches from a fifteen-day routing protocol attack to a denial-of-service attack in a base environment with six troops deployed?

In this experiment, we are considering the difference in how the forces and terrain will perform against two different types of attacks. Military deception has been around for as long as human warfare. This occurs quite frequently in the cyber domain. Attacker forces will start one attack, in order to focus resources on specific terrain, only to then

switch the attack to different terrain. This is the attack vector we are modeling in this experiment. The adversarial force will begin with a routing protocol attack (RPA), and then switch to a denial of service (DOS) attack halfway through the deployment time frame. The following figure shows the change in compromise rate of type 1 and type 2 systems of one run of the virtual experiment. The table below shows the average compromise rate of the type 1 and type 2 systems, after all virtual experiment runs.

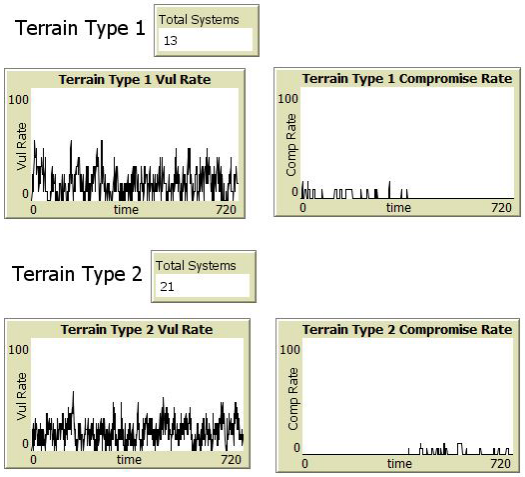


Figure 8: Cyber-FIT dashboard view of virtual experiment

Summary of Simulations	
Number of Forces	6
Environment	Base
Terrain Architecture	Three Tier Distribution
Compromise Rate of Type 1 Systems	1.24
Compromise Rate of Type 2 Systems	0.89

Table 10: Results of virtual experiment

The importance of visualization is displayed in the figure above. The Cyber-FIT interface displays real-time feedback to the user showing exactly what is occurring on the terrain at every time interval. This aids planners and researchers by allowing them to carry out test runs and ensure what they have conceived, conceptually, matches what the model is providing. This shows that under the model conditions, the terrain will hold up quite well against both attacks. The terrain and number of forces deployed, in the base environment will handle a DOS attack better than an RPA. This means that planners and enterprise architects can address this difference. If the difference isn't acceptable,

leadership could send additional resources to the type 1 systems in the way of additional forces or a better maintenance schedule, to decrease the expected compromise rate.

2.1.4.3 What number of forces maximizes expected cyber terrain mission capability rate against random attacks in a tactical environment?

In this experiment, we are considering a tactical deployment and attempting to determine which number of forces maximizes the mission capability rate when the adversary is launching random attacks against the cyber terrain. In this context, mission capability rate is defined as number of cyber terrain agents available divided by total number of cyber terrain agents. When military planners are considering what resources to send to battle, they will attempt to package forces and equipment that will perform at a high level. Since resources are limited, a challenging part of their job is deciding which number of forces will maximize the likelihood that each unit will accomplish its mission. For this experiment, we are modeling a situation where the planners are considering a deployment of cyber terrain which will likely be attacked in multiple ways. So, we selected random cyber attacks for the adversary. Then, we simulated cyber battles against the terrain, each time increasing the number of forces. The following figure shows the results of the simulations.

Figure 9: Mission capability rate as number of forces is increased

As shown, the projected mission capability rate will increase sharply as forces are added. A force package of six troops should provide a mission capability rate above 98.0%. A force package of ten troops should eclipse a 99.0% mission capability rate. The highest number of troops deployed for this set of experiments was 15, resulting in an average mission capability rate of 99.55%. This information would prove quite valuable for determining the appropriate number of troops to deploy to this type of mission.

2.1.5 Discussion

The Cyber-FIT simulation framework, in current form, presents a successful proof of concept by allowing feasible experiments to be crafted and run. The three elements of the model (forces, interactions, and terrain) are all conceptual at this time. Forces differ in vulnerability patching routines, and attack targets. Further development of forces could include: skill level, specialty, experience, and organizational behavior. Terrains differ in types of systems present, vulnerability state, and environmental deployment. Further development of terrain could include: increasing types of systems, realistic lists of vulnerabilities, cost, and access control.

There are nearly limitless potential extensions to this work. For example, in future work we plan to explore various improved definitions of mission capability rate. To define that we'll model various units that depend on different parts of the terrain for mission success. Mission capability rate will be defined as the ability to provide working systems, when demanded. Multiple units could be modeled simultaneously, much like real world operations, which would then provide different mission capability rates for different units at any time. Another example would be adding different types of adversary complexities. Hacktivist organizations, organized crime rings, and nation states would all have different adversarial capabilities and limitations. Then the simulation could predict performance of the forces and terrain against different classes of adversaries

2.2 Cyber-FIT version 2

Cyber-FIT version 2 sought to increase complexity by adding empirical data to a specific behavior found within the model. The candidates for this behavior are the agent classes of terrain, defender, and attacker. Militaries the world over are now operating under the assumption that cyberspace is a contested domain. In an interview, Lt. Gen Bruce T. Crawford, US Army Chief Information Officer, said, "The bottom line, when it comes to the threat, is that never again will we have the luxury of operating in uncontested space. That's become a part of who we are now" [35]. The new reality of contested cyberspace has spawned a new strategy, one that has been gaining traction over the last several years: active cyber defense. Active cyber defense, from a Cyber-FIT perspective, means adding attacker agent complexity, to learn about what the defending agents ought to do.

Denning starts with an active air defense definition, applies it to cyber, and argues that "Active Cyber Defense is direct defensive action taken to destroy, nullify, or reduce the effectiveness of cyber threats against friendly forces and assets" [36]. Dewar describes active cyber defense as part of a triptych that exists alongside fortified cyber defense and resilient cyber defense [37]. In this work, a cross-disciplinary approach is applied in order to analyze the elements of active cyber defense that can disrupt an adversary's attempts at exploiting cyber assets. In order to implement an effective active cyber defense strategy,

an organization must understand which factors are most impactful. Jasper describes the goal of active cyber defense in this way: "detection, verification, and remediation of malicious behavior in the cyber-kill chain, before harm or damage from the breach occurs" [38]. This is a key distinction for a model: the defending agents must detect malicious behavior before the damage can be done, which implies the goal of slowing the adversary down.

2.2.1 Background

The current state of cybersecurity is static and ineffective in managing sophisticated and well- coordinated cyberattacks. Adversarial methodologies are constantly evolving, and the current reactive cybersecurity paradigm is problematic as it manages incidents well after the damage (data/intellectual property theft, system manipulation, or service/functionality disruption) has occurred. Furthermore, this reactive approach gives adversaries more time to get around any security measures set in place and to delve deeper inside systems. Also, eradicating any established adversarial footholds after the cyberattack is costly in terms of manpower and time. Defenders need to stay abreast of this increasingly complex cyber-adversarial landscape, which includes nation-state actors, organized crime groups, and cybercriminal networks. These adversaries use advanced and complex tactics in a persistent manner. Today's cyber adversaries move quickly and change tactics rapidly, which renders the existing reactive cybersecurity paradigm insufficient. Thus, anticipatory cybersecurity measures that identify adversarial behavior and movements are essential; this requires comprehending the human agents conducting cyberattacks, in terms of how they make decisions and how they adapt. This focus on the human dimension of cyberattacks is often minimized in the technical domain.

LeMay et al. proposed [39]the ADVISE method to create executable state-based security models of systems and simulated attack behavior. The method provides probability distributions of attack successes and failures, which can aid security professionals with quantitative security assessments of enterprises. Winterrose and Carter created a set of tools [40] that encode attacker strategies as binary chromosomes that can evolve over time. This strategy simulates the realistic nature of attackers and defenders changing course as they receive new information. Their study replicated attackers knowing which systems defenders would utilize before actually using them, through generational learning, which maps to real-world attacker behavior of conducting reconnaissance before actually attacking systems. Cyber-FIT version 2 aims to simulate attacker reconnaissance activities as well. Cayirci and Ghergherehchi [41] modeled the effects of cyberattacks on decision processes by proposing a series of equations that could be used to simulate human responses. Their equations could be explored using the virtual experiment methodology detailed here. Reed et al. developed a model [42] that simulated the threat responses and workflow of a typical Cyber Security Incident Response Team. Their method included

assigning experience levels to the simulated team members, which differs from this version of Cyber-FIT where attackers and defenders do not have those characteristics. Heckman et al. ran a war game [43] in order to observe the efficacy of denial and deception operations as the primary active cyber defense strategy. Other strategies can be tested in much the same way, informing the collective understanding of active cyber defense. For example, Heydari proposed a Moving Target Defense strategy [44] and showed that it can be successful in hiding IP addresses, thus rendering remote cyberattacks more difficult. Moskal, Yang, and Kuhl proposed a model [45] that simulates various types of attackers against different network system configurations in order to show various ways to penetrate networks. This method is the most similar to the one proposed in this model in that it forces the adversary to step through the cyber-kill chain and conducts virtual experiments to assess overall security posture. The approach described in this chapter differs in that it alters the exploitation assumptions and numbers of defenders to simulate timing of adversaries through each attack phase.

2.2.2 Using adversary behavior from cybersecurity exercises

Real-time cybersecurity exercises provide an ideal platform for studying adversary-defender interactions. The Merit Network and the Michigan Cyber Range provide a robust virtual environment for cybersecurity exercises called Alphaville. Alphaville consists of five 'locations': a school, a library, a city hall, a small business with a manufacturing facility, and a power company. Each of these locations has servers and firewalls with intentional vulnerabilities. A research team attended the 2015 North American International Cyber Summit (NAICS), which hosted a force-on-force exercise in which five teams battled to claim Alphaville's network. The research team observed one competition team of four members over the four-hour exercise, during which they recorded and time-stamped the actions of the team members [46]. The researchers then categorized these actions into different cyberattack (intrusion chain) stages [47]. Furthermore, the researchers also observed key moments of decision-making, facing hurdles, and corresponding adaptations to best capture dynamic aspects of human behavior. The authors recognize that this exercise is not a perfect representative of reality because it occurred in a compressed and expedited manner. However, it provided the researchers with a means to observe human behavior, decision-making, and adaptation as the cyberattack exercise unfolded. More importantly, this time-stamped, categorized data served as actual human-behavior input to agent-based modelling.

2.2.3 Adding empirical data to Cyber-FIT

The Cyber-FIT model is improved by forcing the attackers to move through the cyber-kill chain, based on timing that was observed at Alphaville. The table below outlines the Alphaville observed data and converts those twelve steps to a simpler six-phase cyber-

kill chain [47]. This change was made for two reasons. First, it simplifies the model in by making observed differences in virtual experiments easier to spot. Second, some of the steps in the Alphaville data set were not observed. By converting to the six-phase cyber-kill chain, all six phases could be observed. The reason to make this change in the Cyber-FIT model is to find parameters most likely to slow down the adversary as it moves through the cyber kill chain. By slowing down or forcing the adversary to repeat steps, the defenders have more time to mitigate compromises, discover vulnerabilities, and patch key cyber terrain.

The table below shows the details of the conversion from the researchers twelve step intrusion chain to the six-phase cyber kill chain. As noted in the last column of the table, if a system is in a vulnerable state, only two of the six phases would be affected. In other words, in order for an attacker to successfully complete the recon, weaponization, command and control, and actions on objectives phases, no system vulnerability need be present. Therefore, the goal is to determine what defensive behaviors will stall the adversary in the other phases that do depend on a vulnerable system state: delivery and exploitation

Step	Name	Minutes	Phase	Name	Minutes	Vulnerability Affects?
1	Define Target	0	1	Recon	75	No
2	Organize Accomplices	55				
3	Build/Acquire Tools	20				
4	Research Target Infrastructure	25	2	Weaponization	50	No
5	Test for Detection	25				
6	Deployment	20	3	Delivery	20	Yes
7	Initial Intrusion	35	4	Exploitation	35	Yes
8	Outbound Connection Initiated	20	5	Command and Control	20	No
9	Expand Access and Obtain	60	6	Actions on Objectives	85	No

	Credentials					
10	Strengthen Foothold	25				
11	Exfiltrate Data	0				
12	Cover Tracks	0				
Total		285	Total		285	

Table 11: Converting twelve-step intrusion chain data to six-phase cyber kill chain

2.2.4 Virtual experiments

After adding the adversary intrusion chain behavior to the Cyber-FIT model, five virtual experiments were conducted. The first experiment set out to ensure the model behaves similarly to the empirical data, in terms of both average time per phase and range of times observed. The next four experiments altered agent rulesets and environment variables in order to examine assumptions, conduct what-if analysis, and inspect simulation results. Since only phases three and four depend on the presence of vulnerable terrain, the final four virtual experiments will primarily focus on the phase-three and phase-four effects. For each experiment, the motivation will be explained, results presented, and implications discussed from a military organizational perspective. For each virtual experiment test case, all three attacker agents attempt the same exact attack, each traversing through the cyber-kill chain independently of each other. The table below displays the independent and dependent variables of interest for all five virtual experiments

Independent Variables	**Variants**	**Values**
Attack Type	3	DOS, RPA, Phishing
DCO Forces	20	1, 2, 3, … 20
Exploit Success Rate	20	.02, .04, .06, … .40
Phishing Attack Targets	30	3, 6, 9, … 90
Vulnerability Growth Rate	19	.01, .02, .03 … .1, .2, .3, …1
Dependent Variables	**Variable Type**	
Phase Completion Time	Continuous	
Terrain Agents Compromised	Integer	

Table 12: Independent and dependent variables for virtual experiments

2.2.4.1 Virtual experiment one

In the first experiment, the model is tested to ensure that the attackers move through the cyber-kill chain in accordance with the empirical data. This is done by controlling the attack type variable to denial of service and holding exploit success rate at 15%. Then, number of DCO forces is altered from 0, 2, 4, 6, and 8. The table below shows the results of the simulations.

Phase	Empirically Observed Time	Average Time Simulated	Range Simulated
1	75	79.08	[61 - 141]
2	50	55.95	[36 – 156]
3	20	144.58	[10 – 2,281]
4	35	48.37	[5 – 589]
5	20	24.17	[6 – 105]
6	85	88.88	[71 – 138]

Table 13: Virtual experiment one results

As shown in the table, the model is behaving according to the empirically observed data. The empirically observed time falls in the range simulated for each phase. Also, as expected, the average time is close to the empirical time for the four phases (1, 2, 5, 6) that do not depend on vulnerable systems. The two phases (3, 4) that do depend on vulnerable

systems have a much larger range. As DCO forces are added, it is more likely that system vulnerabilities are removed, which causes delivery to be delayed (phase three) and exploitation to become impossible (phase four). Also, if at any time during phase four, the delivered payload is successfully defended and mitigated, the attacker moves back to phase three to attempt delivery on a different terrain system. Since the model is behaving as expected, the next four virtual experiments can be performed with confidence that this model has been output validated. For the next experiments, the experiments examine different aspects of the model that will most affect and be concerned with phase-three and phase-four responses. These are the two phases that can most likely be slowed down by improved active cyber defense.

2.2.4.2 **Virtual experiment two**

For the second experiment, the research question is: 'how many DCO forces should be deployed to maximize the time to complete phases three and four during a routing protocol attack with an exploitation success rate of 15 percent'? This is a continuation of previous work detailed in the previous section, but with more complex behavior. That previous virtual experiment addressed the following question: "How many forces should we deploy to minimize the effect of a routing protocol attack in an industrial environment?" Results showed that deploying eleven DCO forces would be appropriate because any increase above eleven was within one standard deviation of previous force levels. With limited numbers of troops, commanders must decide how best to deploy force packages. In this experiment the research question has the same spirit but is altered along the lines of the added behavioral data. How does increasing DCO forces affect expected time for attackers to complete phases three and four, on average. The figure below shows the results of this virtual experiment.

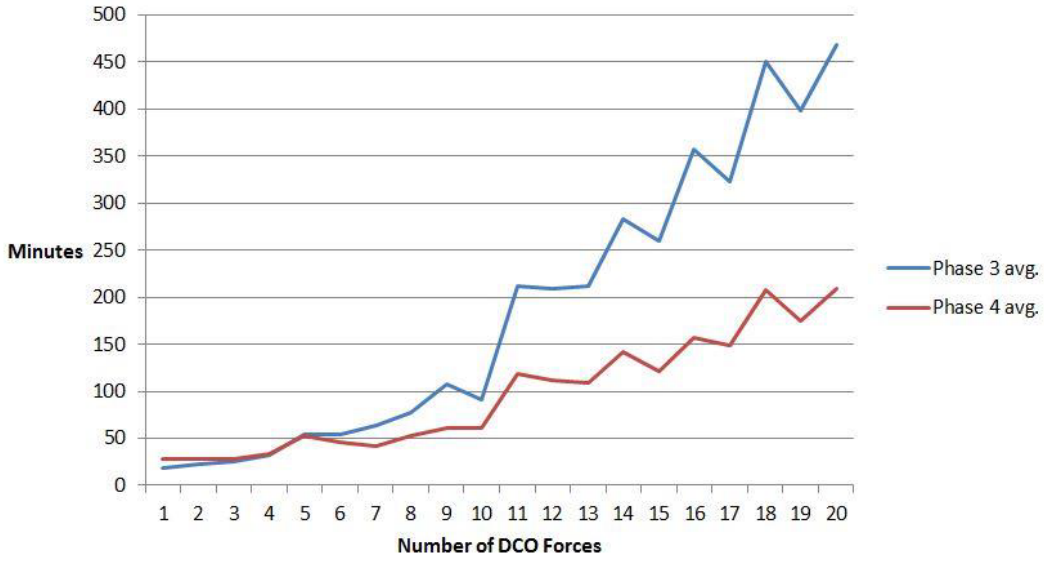


Figure 10: Virtual experiment results

As shown in the figure above, increasing the number of forces results in time to complete phases three and four power curves. Interestingly, it seems that approximately eleven DCO forces may be the appropriate number of troops to deploy given these circumstances and assumptions, as was the case in the previous work. On average, ten DCO forces caused the phase-three completion time to be 91.37 minutes. Eleven DCO forces resulted in an average phase-three completion time of 211.4 minutes, an increase of 131.37% over ten DCO forces. A similar increase in phase-four completion time also occurred at eleven DCO forces, with an increase of 97.4% above ten DCO forces. Thus, deploying DCO force packages can have dire consequences for military organizations because there are quite simply not enough troops to deal with the ever-increasing dependence of all missions on cyber terrain.

2.2.4.3 **Virtual experiment three**

In this experiment, exploitation success rate was held constant at 15 percent, along with four defending force agents. The question being addressed in this virtual experiment is: 'how many user systems will be compromised as phishing attack targets are increased'? Each simulation run added three user- system targets, starting with three and ending with 90, to the attack list for each of the three attacker agents. In real-world operations, adversary organizations would vary the number of phishing targets from low (less likely to be detected) to high (greater likelihood of users clicking through). For this experiment, the researchers mimicked that behavior and observed the emergent behavior across the entire spectrum of user-system attack targets. The figures below display the results of this virtual experiment.

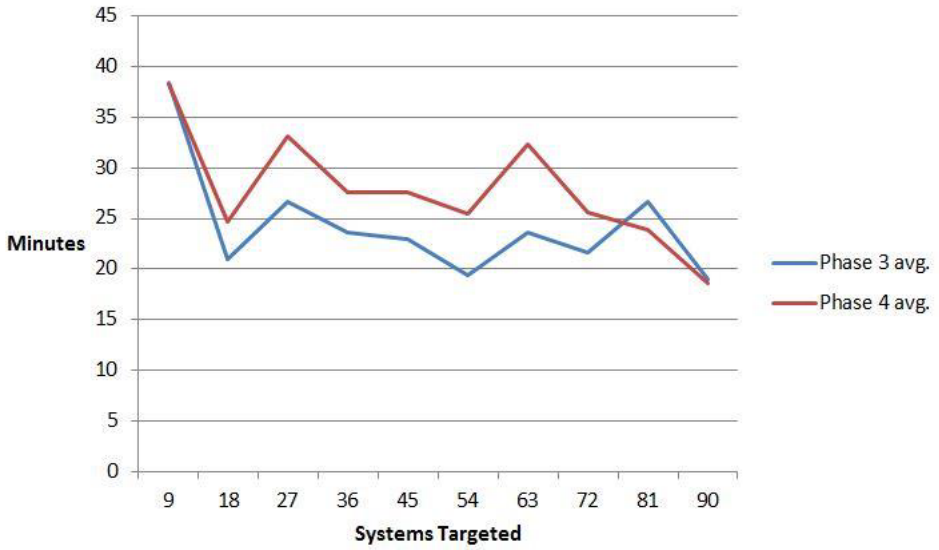


Figure 11: Phase 3 and Phase 4 average time to complete for virtual experiment

Figure 12: Average number of systems exploited result of virtual experiment

As shown in Figure 11 above, as the number of systems targeted increases, the time required to complete phases three and four decreases, as expected, since there are more attacks for the defending agents to work through. The average total time to complete phases three and four decreases by 50.87% when increasing attack targets from nine to 90. Interestingly, even though the attacker is moving more quickly through the cyber-kill chain, the number of systems compromised also increases. When each attacker targets three user systems, 4.8 systems on average are exploited, or 15.48% of all user systems. When each attacker targets 30 user systems, 15.4 systems on average are exploited, or

49.68% of all user systems. This experiment was run at exploitation success rate of 15%, for each time tick (one simulated minute). Military organizations could test assumptions about how likely their user base would be to click through phishing emails, given various forms of training and policy. Then, they could weigh options based on cost and the disruption each would cause as compared with expected decrease of user- system compromise, given future phishing attack predictions.

2.2.4.4 **Virtual experiment four**

In this virtual experiment, the research question is: 'what is the average time to complete a routing protocol attack with eight DCO forces deployed, as the vulnerability growth rate increases'? Vulnerability growth rate is the percent chance that a vulnerability appears at any give time tick. Cyber terrain, that is up to date, regularly patched, and monitored, will have lower vulnerability growth rate than systems not regularly maintained. In this experiment, the vulnerability growth rate is altered in order to observe the impact this variable has on time necessary for the adversary to traverse the intrusion chain. Clearly, the more vulnerabilities present on cyber terrain, the more likely that attackers will be successful. In the Cyber-FIT model, at any given time, a vulnerability may appear on terrain based on the vulnerability growth rate. Military organizations pay close attention to system vulnerabilities and strive to minimize vulnerabilities across the enterprise, which is being simulated using the vulnerability-growth-rate agent variable. The figure below shows the results of the first run of simulations, varying vulnerability growth rate from 1% to 100% by intervals of 10% in a routing protocol attack with eight defensive force agents deployed.

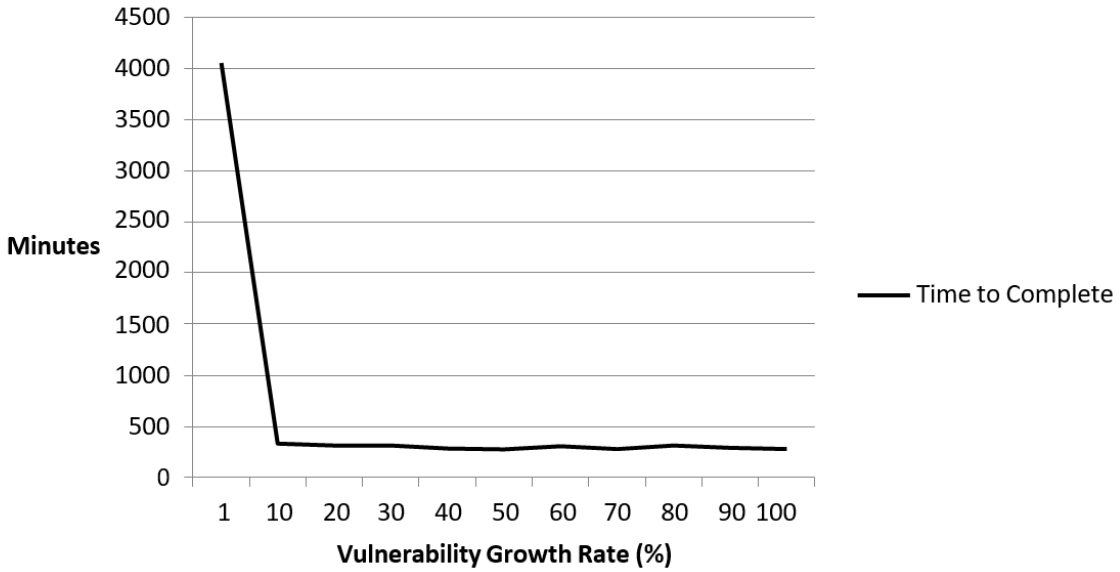


Figure 13: Results of virtual experiment, part one

As shown in Figure 13, at 1% vulnerability growth rate, the average time to complete a routing protocol attack with eight defensive force agents deployed is over 4,000 simulated minutes. But at 10% vulnerability growth rate, the number drops to 333. Increasing the vulnerability growth rate further has little to no effect on expected time for the adversary to complete the attack. Based on these results, the simulation was re-run but this time only examining the response surface resulting for values between one and ten percent. The figure below shows the resultsof the continuation of virtual experiment four.

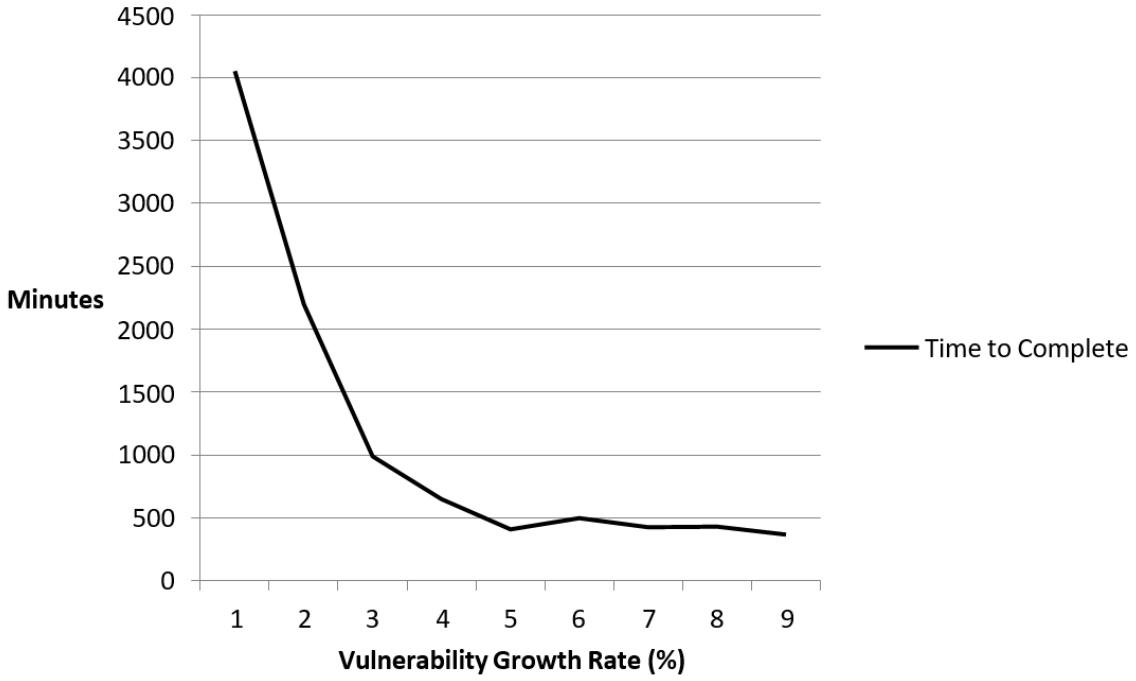


Figure 14: Results of virtual experiment

As shown in Figure 14, a more granular view of the data can be very powerful. There is a precipitous drop from 1% to 3%, a gradual drop from 3% to 5%, and then no substantial effect from that point forward. This means that, given the assumptions embedded in this virtual experiment, organizations should take every effort to keep the vulnerability growth rate as low as possible. By taking efforts to move the vulnerability growth rate from 3% to 1%, the expected time to complete a routing protocol attack moves from 990 minutes to 4,053 minutes, or a 309% increase. This gives defensive cyber forces a much greater chance of recognizing and then mitigating the attack.

2.2.4.5 **Virtual experiment five**

In this virtual experiment, the research the question is: 'What is the expected time to complete phases three and four, during a denial-of-service attack, with six DCO forces deployed, as the exploitation success rate is increased?' One of the most important parameters in the current model is the exploitation success rate. Once the attacker has reached phase four, the model forces five ticks (simulated minutes) to occur before attempting to exploit the systems that have payload delivered. This simulates a mandatory minimum waiting period for an exploit to take. After the waiting period has expired, at every tick, a random number is generated. If the random number is less than the exploitation success rate, the exploit is successful, which means that system has been compromised. Also, at every time tick, a DCO force might discover a system vulnerability and remove the payload that was delivered in the previous phase. In this event, the attacker

would move back to phase three and would re-attempt payload delivery on randomly selected server systems (the target of a denial-of-service attack). It is expected that, as the exploitation success rate increases, the time it takes to traverse phase three and phase four will decrease, the question is: by how much? The figure below shows the results of this virtual experiment.

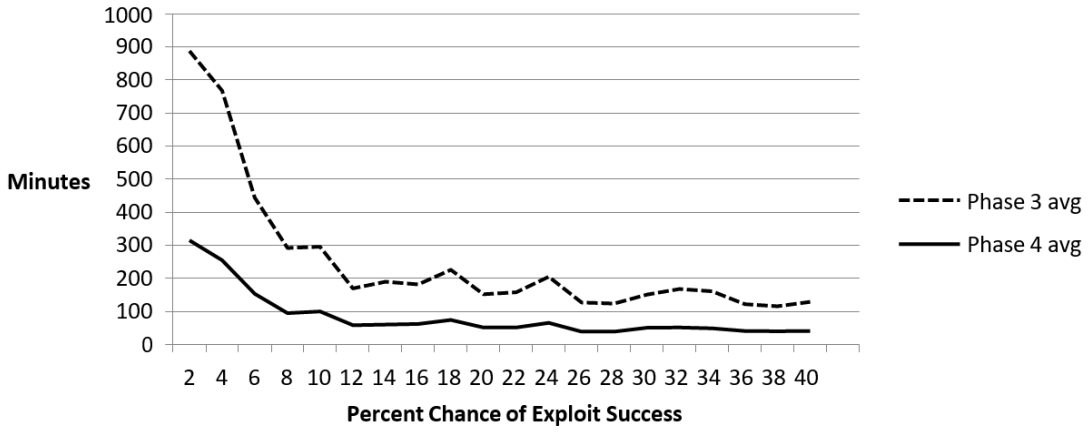


Figure 15: Results of virtual experiment

As shown in Figure 15, the average time taken to complete phases three and four will decrease dramatically as the exploitation success rate increases from 2 to 12, but then has little effect onward. This emergent behavior shows the power of agent-based modelling, as this result would be very hard to predict without virtual experimentation. The implication here is that military organizations should spend significant resources attempting to move the denial-of-service exploitation success rate in the lower range of the interval. That is, if defensive measures can be taken, that decreases the exploitation success rate from six to four, then the organization will increase the expected time for a denial-of-service attack to complete phases three and four by 425.11 minutes, or 70.96%. This gives DCO forces a significantly better chance at removing the denial-of-service attack threat, especially if indicators of compromise are recognized during phase three.

2.2.5 Discussion

By adding the empirically observed adversary intrusion chain behavior data to the Cyber-FIT simulation framework, the model and ensuing virtual experimentation realized an added a level of complexity. This leads to improved experimentation and a more complex what-if analysis. As more data are observed and added to the framework, the model becomes more realistic. In experiment one, the model was shown to be able to simulate the empirical data. Experiment one could be extended by observing more cyber warfare exercises and aggregating many teams' progress through the intrusion chain, which would move closer to realistic phase timings. In experiment two, there is a point at which

DCO forces can have a significant impact on slowing down the attacker's progress through the kill chain. Experiment two could be extended by adding skillsets to the defenders so that they defend terrain that aligns with their training and military occupational specialty, as it happens in real-world operations. In experiment three, there was an increase in the expected number of user-system compromises by varying the phishing attack targets. Experiment three could be extended by adding user-training levels and phishing-complexity levels to find a point at which organizational training minimizes various types of phishing attacks. In experiment four, the vulnerability growth rate above 3% leads to fast movement through the cyber-kill chain for adversaries, which validated the vast resources military organizations spend on minimizing vulnerabilities every way possible. Experiment four could be extended by defining types of vulnerabilities and forcing the adversary to match payload with targeted vulnerability before moving out of the Exploitation Phase. Experiment five showed that exploitation success rate assumptions can have a very large effect on how quickly an adversary is able to traverse the cyber kill chain. Experiment five could be extended by adding observed exploitation success rates across different types of terrain and then varying attacks that target different terrain.

This research shows that a single cybersecurity exercise case study based on actual human behavior data (rather than probabilities) can be subjected to different simulation experiments. This methodological integration of social science cyberspace research and computer science simulation offers new insights into adversarial and defender behavior. It is important to note that the criminological discipline also benefits from such multidisciplinary methodological fusion. Adversarial behavior, decision-making, and adaptability have long been studied in the criminological domain. However, real cyberattacks are difficult to observe as they are covert and fast-paced. Criminologists usually have limited access to good quality cybersecurity exercises. And when criminologists do have access to these exercises, they often cannot control the exercise environment or introduce variables to manipulate the exercise; in short, the criminology researcher can only be a passive observer. Furthermore, effectively analyzing qualitative data, such as a single observed cybersecurity exercise, is time-consuming. Simulations offer a robust mechanism for criminologists to overcome each of these hurdles. Criminologists can work with computer scientists to replicate exercises with different permutations and combinations of scenarios, defender behavior, and adversarial behavior. Thus, the methodological mix of social and computer science not only contributes to the field of cybersecurity, but also to the criminological domain.

2.3 Cyber-FIT version 3

The primary design goal of Cyber-FIT version 3 is to incorporate a theoretical concept into the model. There are many candidates for incorporation. The terrain agents, in real world operations adhere to theoretical principles in some fashion such as network

theory or communication theory in terms of performance and speed. Theories around team-based dynamics like transactive memory and organizational learning could be explored and applied to the force agents with an agent-based model like Cyber-FIT, which was an original goal of the work: the opportunity to extend many types of functions onto the basic model architecture. One such functional extension candidate, which has garnered attention in military research circles recently, is "cyber situational awareness".

Military leaders are keenly aware of the pressing need for improved cyber situational awareness capabilities. In recent testimony to the U.S. Senate Armed Services Committee [48], Navy Vice Admiral Michael Gilday said: "We've extended our defensive posture to include deploying defensive cyber teams with our carrier strike groups and our amphibious readiness groups". This means that an ever-growing number of military operations will have a defensive cyber force attached. When defensive cyber forces fall into an area of responsibility, they must first conduct a survey of the cyber terrain, like an infantry unit would survey the land terrain, or an air controller would examine the air space. Militaries have been conducting land terrain surveys for thousands of years, but cyber terrain surveys for less than a decade. In this version of Cyber-FIT, a cyber terrain survey mission can be simulated where defensive force agents keep track of cyber terrain agents with a cognitive computational model. It is show that given several realistic behaviors and constraints, the defensive cyber force can conduct a full survey in approximately two hours, but full cyber situational awareness may be impossible. A visual representation of cyber situational awareness is depicted by the U.S. Army in the figure below. Ultimately, this is a picture of how data are defined and transmitted throughout the network of nodes and links under the Army's control.

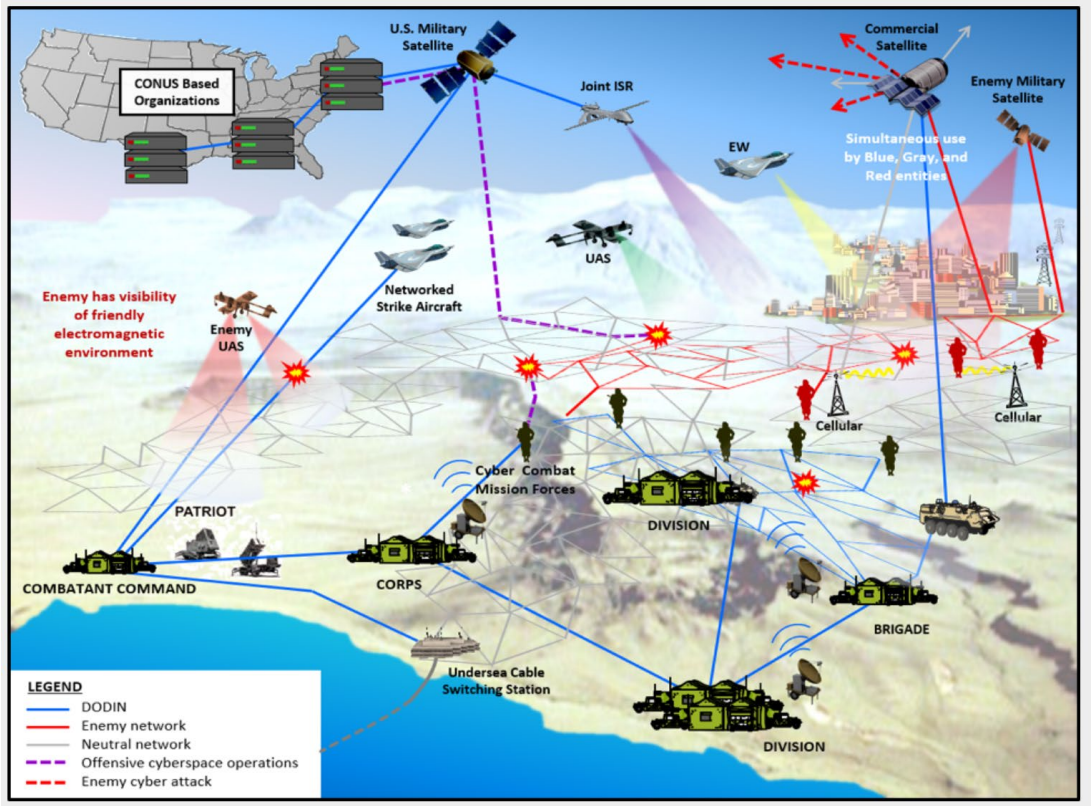


Figure 16: U.S. Army notional depiction of cyber situation awareness [49]

The purpose of surveying cyber terrain, whether on a corporate network, or military mission, is to gain understanding of the states of the various systems under the team's purview. Put another way, it is to gain "cyber situational awareness". There are many definitions of cyber situational awareness. Onwubiko [50] defines cyber situational awareness as "processes and technology required to gain awareness of historic, current, and impending (future) situations in cyber". In this model version we are modelling the knowledge of the current situation that the defender has realized. Similarly, Barford et al. [51] describe seven aspects of cyber situational awareness. The first is "Be aware of the situation. This aspect can also be called situation perception". Cyber-FIT simulation software defines the perception of the agents' knowledge of the terrain as a table of system states. This gives a computational model of the cognitive representation of cyber situational awareness for each agent, and cumulatively, for the team. As time goes on, they build and update a table of terrain states. The states are one of three: not vulnerable, vulnerable, and compromised. This is compared against the true state of the systems at every minute togive the team Cyber Situational Awareness (CSA). That is,at every time tick (one simulated minute), the agents store the value of the state of the system they are interacting with. This models their cognitive understanding of the cyber terrain. The team's cyber situational awareness is the sum of their cognitive models. At time 0, they have no cyber situational awareness. By defining cyber situational awareness in this manner, we

can observe the changes over time, determine what factors most affect it, and more clearly understand what the appropriate definition of cyber situational awareness is, in a given scenario.

2.3.1 Virtual experiments

We conducted two virtual experiments using this model. In each experiment, we hold the attack type, number of agents, vulnerability growth rate, exploit success rate, and defensive action success rate all constant. Those variables can be altered to explore the response of the model but is not necessary for these two experiments. In these two experiments we are examining how successful the terrain survey is, as defined by team level cyber situational awareness and how quickly the agents can survey.

2.3.1.1 Virtual experiment one

In this virtual experiment, the research question is: What is the maximum cyber situation awareness during a cyber team survey? This experiment simulates a defensive cyber force falling into contested cyber terrain under active attack. Three defensive agents survey and defend the terrain, while three offensive agents attack the terrain. The goal of this experiment is to determine how successful the survey is, and how much time should surpass until the performance levels off. As shown in the figures below, the performance levels off after 100 simulated minutes, but there is still a fair amount of variance between runs of the experiment. After 100 minutes, the minimum CSA observed was 0.40, the maximum was 0.86, and the average was 0.64.

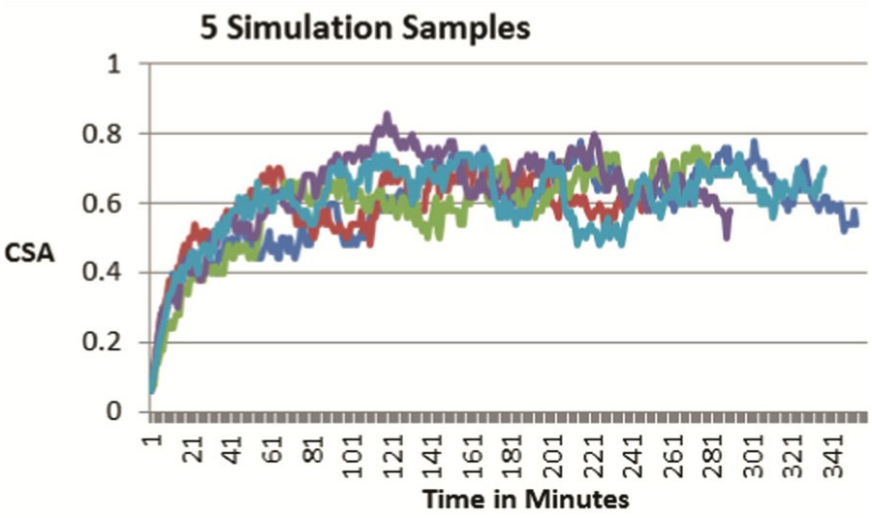


Figure 17: Virtual experiment one simulation samples

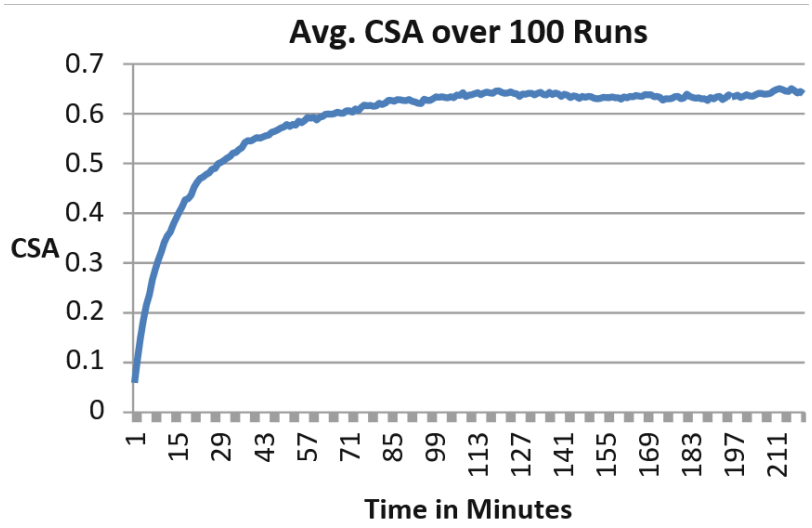


Figure 18: Average Cyber Situation Awareness over 100 runs

2.3.1.2 Virtual experiment two

In this experiment, we are running the simulation until the agents' cognitive model of cyber situational awareness covers all 50 cyber terrain points. In this experiment, we observed that the average time to complete the full survey (all 50 cyber terrain endpoints inspected) was 115.81 min and average CSA at that point was 0.64. As shown in the figure below of a scatter plot of results, CSA does not improve when agents take longer to complete the survey mission.

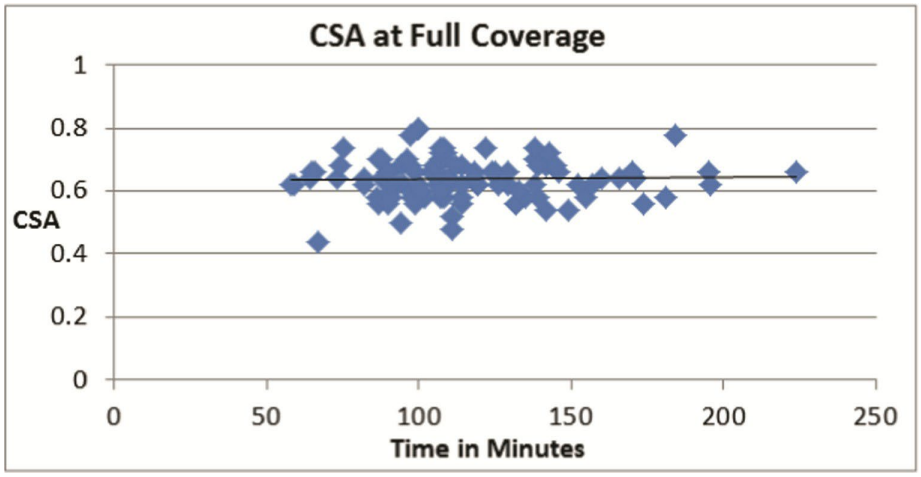


Figure 19: Cyber situation awareness change over time

2.3.2 Discussion

The key finding of experiment one was that over 100 runs, the maximum CSA observed was 0.86. This is expected because agents can only inspect one piece of terrain per minute. Like in real life, as one system is being inspected, other systems may become vulnerable or compromised. Military leaders must decide how many resources to apply to a cyber terrain survey, and how much risk they will accept, given the fact that 100% cyber situational awareness is impossible. Also, when vulnerabilities are found, which should be immediately elevated, which should be immediately fixed, and which can be left to fix later? Also in experiment one, after minute mark 100, at any given time, the CSA ranged from 0.40 to 0.86. This is a fairly large performance gap. Cyber forces should consider defining what routines and processes increase the likelihood of higher cyber situational awareness.

In experiment two, we found that the average time to conduct a full survey is 115 minutes. This is based on the agents randomly selecting terrain and switching every minute. In an operational mission, military leaders should develop detailed cyber terrain survey plans, with clear reporting instructions. This will ensure that survey missions are repeatable and measurable. Also, careful attention should be paid to the order in which terrain is surveyed. In this simulation, order of operations does not matter, which would not be the case in an operational environment.

This version of the model was able to incorporate the functionality necessary to explore a theoretical model. This was an approach to computationally defining team cyber situational awareness as an accumulation of the agents' cognitive model of the state of cyber terrain, compared with the true state of the cyber terrain. We conducted two virtual experiments to assess the assumptions of the model and reason about the applicability of the findings. This work is part an ongoing effort to improve the state of the art of military based cyber force package modelling and simulation. In future work we plan to simulate the passing of messages between agents, in order to share cyber situation awareness, and collectively act upon cyber terrain vulnerabilities and threats. Also, we'll create simulations where more realistic constraints are applied, which will force the simulated commander to make resource trade off decisions.

3 Chapter 3: The Performance Measures of Cyber Teams

As long as militaries have been organized, military training is a key component to hone the skills necessary for success. The U.S. military takes training very seriously and spends a substantial amount of time and resources on it. For example, in 2018, the Army requested [26] $429 million in funding for a new cyber training range capability. The cyber range will have capabilities to simulate "real-world mission rehearsal" and necessary friendly, supporting and adversarial forces. Many of these concepts are quite clear to the developers and personnel building the cyber range systems. What is not clear, is how cyber mission forces are assessed. On September 26, 2018, at the Joint hearing to receive testimony on the cyber operational readiness of the Department of Defense [27], Brigadier General Dennis Crall, principal deputy cyber advisor at the Office of the Secretary of Defense said the following: "I would say we need to ensure that we have a solid baseline and assessment mechanism so, when we come back here and talk to you about what's working and what's not working and how we've spent money, we can do so with the right kind of accountability". In other words, at that time, there was no baselines mechanism to know how well cyber forces are prepared for their missions. For individual personnel, we do have some semblance of their experience, knowledge, and skills, by simply knowing what education and certifications they've completed. When we aggregate those skills and experience into teams, it is very difficult to compare teams, and predict how well given teams would do in given mission sets. Furthermore, there is no team-based assessment baseline, to understand objectively, who the elite cyber forces are.

This thesis will provide a way to computationally compare teams and run virtual experiments that allow the user to ask what-if questions on how to approach cyber team construction. A capability like this would provide the accountability that lawmakers are requesting, to validate the expenditure of such large resources. The primary contribution of this work is in the metrics and methodology to computationally model cyber team performance. Brigadier General Crall stated the need for a "baseline and assessment mechanism". That can only be done by defining the metrics and measures of cyber teams, and then laying out a systematic and repeatable mechanism with which to compute them.

The key question that will be addressed in this thesis is: How well is the cyber team performing? Consider the following scenario: a cyber team is conducting a hunt mission due to a network breach, searching for malware. On day three, the commanding general walks into the watch floor and asks: "How is the team doing?" Determining the answer to this seemingly simple question becomes very complicated, very quickly. Should we look at their network anomaly reports and check for accuracy? Should we check the status of key systems, and if they are reporting up and running, assume the team is doing fine? Should we ask how they think they are doing? Answering these questions turns out to be quite difficult. It remains very difficult to determine the efficacy of cyber operations, and, by extension, how well a cyber team is performing at any given time. Currently, there is

no systematic way to determine how well a cyber team is performing, or compare teams against each other. Most often, team leadership makes expert assessments with little in the way of evidentiary data backing them up. One could say the assessment of cyber team performance is much more qualitative and abstract, than quantitative and methodical. This work intends to address that gap. As compared to other types of teams, cyber teams have the advantage of almost constantly interacting with systems that log actions being taken. This means that data is available which could be collected, and then processed through algorithms that provide performance indicators. A description of the proposed measures of cyber teams is provided in the table below.

The primary design goal of this version of the Cyber-FIT Simulation Framework is to provide an apparatus to comprehensively and quantitatively measure the performance of a cyber team. This means after each run of the simulation all data is present in the output files that can be used to measure the simulated team performance, answering the question: How well did the team do? In order to create a list of performance measures, many conversations with subject matter experts have occurred. These conversations have occurred at cyber war exercises, cyber war-gaming sessions, cyber operations doctrine writing sessions, cyber security and simulation conferences, and through work at Carnegie Mellon University through CASOS Center events and workshops. Finally, a focus session was spent with a diverse group of military cyber operations planning experts validating the current model design and behaviors that lead to the collected data performance measures. The rest of this section is a detailed description of each measure. Each measure is explained as to how it is calculated, what behaviors affect the measures, what control variables affect the measure, and how this measure could be collected in operational systems. The following table defines all model specific terms referenced frequently throughout the remainder of this section.

Term	Description
Tick	A simulated time unit or period. Typically in agent-based modeling each tick represents a second, minute, hour, or other user defined time period.
Cyber Team	For the entirety of this technical report, cyber team refers to a group of defender agents assigned to the same team. Cyber-FIT allows for multiple team simulations but each performance measure is specific to a cyber team of defender agents.
Mission Defined	This refers to variables that are user defined in mission configuration files and context dependent. Mission defined can refer to expected mission outcomes, timing consideration, and details defining kinetic and friendly forces.

Table 14: Section 3 Common Terms and Descriptions

3.1 Terrain Vulnerability Rate

Terrain vulnerability level represents the total vulnerability level of a given network of computer systems (cyber terrain). From a modeling and simulation software perspective, this is an example of a very specific agent by agent measure that can be computationally quantified and aggregated to total terrain vulnerability level. In the Cyber-FIT model this means each terrain agent has a temporally changing list of vulnerabilities ranging in identification number from 0 – 99. Each identification number is also a proxy value representing the severity level of the vulnerability. The higher the number the more vulnerable this particular vulnerability makes the terrain agent. So, a terrain agent with vulnerabilities 90 and 80 is much more vulnerable than a terrain agent with vulnerabilities 9 and 8. This means the worst possible scenario for one terrain agent is that it becomes vulnerable to all attacks, or its list of vulnerabilities is all integers in the range [0,99] which is 4,950. Summing all vulnerabilities over all terrain agents gives total vulnerability level. Dividing by the number of terrain agents gives the terrain vulnerability rate.

3.1.1 Computation

Define T as the set of all mission terrain agents as a subset of all agents A

$$T \in A$$

Define V as the set of all vulnerabilities, V_i, that terrain agent T_J can have

$V_{T_j,i} \neq 0 \leftrightarrow T_j$ has vulnerability i

Then, total vulnerability level TVL of T is calculated by

$$TVL = \sum_{j,i} V_{T_j,i}$$

Finally, to normalize, TVL is divided by total possible vulnerability level for each terrain agent j, giving terrain vulnerability rate, TVR

$$TVR = \frac{TVL}{4{,}950j}$$

3.1.2 Operational Considerations

This military focused definition is similar to long-standing concepts of terrain-based risk assessment. In conflicts, militaries will analyze, for example, land terrain positioning and determine where and how they are vulnerable to attack. This can be based on geographic considerations such as access to water, proximity to supply chains, difficulties with mountains, etc. In the newly emergent concept of cyber terrain, militaries will similarly conduct vulnerability assessments on this terrain type. Rather than analyzing physical components, the analysis is based on logical components, systems architecture, networking, software, and cyber security. When military cyber teams are deployed to protect networked systems, one of the first artifacts produced is a terrain vulnerability assessment. The assessment will touch upon aspects similar to those just mentioned. Terrain vulnerability level, at face value, is one of the most easily understood performance measures of a cyber team.

The primary purpose of any military or corporate Information Technology (IT) department, is to make the network less vulnerable to attack (minimize terrain vulnerability level). This is done near continuously, every day, through system monitoring and updating. There are many examples of detailed vulnerability data (gleaned from enterprise security tools) being aggregated and utilized for advanced securtiy techniques [52]. Most IT offices will have dashboards displaying vulnerability status of a wide array of systems. Those more vulnerable might be displayed yellow, and active problem systems could be red. Therefore, snapshots of system vulnerability level can be shown throughout the cyber team operations providing real-time quantified values of terrain vulnerability status. This measure is already reported on in real-world operations and arguably the closest to tracking the ground truth.

3.2 Terrain Vulnerability Change

Terrain vulnerability change builds upon the previous section by adding the change to vulnerability level over time. This represents a change measure at any given time period. In the Cyber-FIT model, this is measured by fitting a curve over a given period of ticks and then plotting the derivative of that function.

3.2.1 Computation

As previously defined, terrain vulnerability rate, TVR can be measured over time.

Therefore, define terrain vulnerability change, TVC calculated by

$$TVC = \frac{\Delta}{\Delta t} TVR$$

3.2.2 Operational Considerations

This performance measure is more indicative of mission success in that it is a clear measure of how more or less secure the assigned cyber terrain is, after a period of time has surpassed. Ideally, a cyber team assigned to secure a network of computer systems will cause the terrain vulnerability to decrease, which would be apparent, visually, by graphing and displaying terrain vulnerability change for the duration of the cyber operation. This measure, like terrain vulnerability rate is already regularly used in both military and industry cyber security operations centers and information technology offices. Systems are normally set to alert when a vulnerability rate is detected to change above an abnormal threshold, which essentially means the terrain vulnerability measure has increased too quickly or above a threshold value.

3.3 Terrain Compromise Rate

Terrain compromise rate represents the rate of compromised systems present on the network. This measure is one of the most direct measures of cyber team success, as preventing systems from being compromised is the primary goal. Reducing terrain vulnerability rate reduces the likelihood that terrain might become compromised, but ultimately system compromise is what the team is aiming to prevent. In the Cyber-FIT model, terrain compromise rate is computed by dividing number of terrain agents in a compromised state by total number of terrain agents at any given time in the simulation.

3.3.1 Computation

Define T as the set of all mission terrain agents as a subset of all agents A

$$T \in A$$

Define T_c as the subset set of all T that are in a compromised state

$$T_c \in T$$

Define terrain compromise rate TCR calculated by dividing the absolute value of the compromised set by the absolute value of the full set

$$TCR = \frac{|T_c|}{|T|}$$

3.3.2 Operational Considerations

The ideal state for any cyber team focused on securing an operational network is to have zero compromised systems. However, over a long enough time period some systems will inevitably become compromised, even if through non-malicious means. A system that is simply "down" due to outdated software, hardware failure, user error, system interruption, power issues, etc., will still likely be considered compromised, at least initially from an incident response perspective. With a large enough network, considering compromised systems that have become inoperable for unknown reasons, a compromise rate above zero is inevitable. This is another measure regularly known to real world operation centers at the current time. The state of technology already allows for the tracking, reporting, and analysis of this measure. Security information event management (SIEM) solutions are widely used in military and industry organizations, tracking system responses from health checks. Unresponsive systems are identified and alerts are sent to analysts. Tracking this measure over time is already built into SIEM capabilities. There is much research using widely available intrusion protection systems for instance event log trend analysis [53] and differences in behavior by infected versus non-infected hosts [54].

3.4 Terrain Compromise Rate Change

Terrain compromise rate change builds upon the previous section and represents how terrain compromise rate is changing over time. In the Cyber-FIT model this means a curve is fit plotting terrain compromise rate over ticks and taking the derivative at every tick.

3.4.1 Computation

As previously defined, terrain compromise rate, TCR, can be measured over time

Therefore, define terrain compromise rate change $TCRC$ calculated by

$$TCRC = \frac{\Delta}{\Delta t} TCR$$

3.4.2 Operational Considerations

This measure is similar to terrain vulnerability change as it is very well understood as an indicator of a successful military cyber mission, or period of time in an industry cyber security operations center. Terrain compromise rate is operational as of now in that organizations are closely monitoring system availability. Clearly, if the compromise rate decreases over time, the teams are performing well and the organization has a more secure posture. This measure is different than terrain vulnerability change in that it is much more challenging to measure. This is for the simple reason that SIEMs are much better at defining specific vulnerabilities, due to the industry-wide work that goes into vulnerability identification. Compromises are more difficult to define. However, if an organization assumes some noise will follow the compromise system signal, then there should be a pattern and moderate regularity to the noise. For example, if some number of systems per year appear down, due to a hardware failure, then that network behavior should fall into a steady state. The important consideration for actually measuring terrain compromise rate change is to keep track of all down systems over time, and visually manage. Performance dashboards tracking the terrain compromise rate change historically would be vital in order to know if the current state is better or worse.

3.5 Mission Compromise Time

Compromise time is a measure of how long computer systems are compromised before the cyber team can restore them. In the Cyber-FIT model, this means that a terrain agent has changed state to compromised due to a successful attack by an attacker agent. The time from state changing to compromised, until a defender agent becomes aware of the compromise and then restores the terrain agent to normal, is compromise time for that particular terrain agent. The total time amongst all terrain agents in a compromise state is compromise time for a given campaign simulation.

3.5.1 Computation

Define T as the set of all mission terrain agents as a subset of all agents A

$$T \in A$$

For each mission terrain agent T_i, define c_i as the cumulative compromise time

Define mission compromise time MCT as the sum of all cumulative compromise time over all mission terrain agents

$$MCT = \sum_i c_i$$

3.5.2 Operational Considerations

Compromise time is clearly an important measure of cyber team performance. The longer systems are compromised, the longer the attackers have to complete their own objectives. Usually, these objectives include lateral movement within the network, exfiltrating data, causing damage to systems that result in other software or hardware-controlled failures. Therefore, a well performing cyber team should be able to first recognize when systems are compromised, and then restore those systems in a timely manner. This measure follows along with the previous sections' discussion. Determining when a compromise occurs is still very difficult, due to the advanced persistent threats present on real world systems all over the world. Similar to compromise rate change, if an organization is tracking down time for any systems, and visually graphing the metrics around that, they can begin to understand normal trends within their networks. This measure is not fully operational at this time. This is due to the difficulty with attributing a down system to a known malicious actor. Systems can be recognized by SIEMs to be down. However, those downed systems could be not responsive due to a network issue, for example, that has nothing to do with a malicious attack. A more realistic measure of mission compromise time would be achievable by adding a more detailed capability. For instance, the SIEM shows a down system, then a tool interrogates that host running an automated diagnostics security check.

3.6 Time to Detect

Time to detect refers to the amount of time it takes for a cyber team to recognize a system has been compromised. In the Cyber-FIT model, this means a defender agent has interacted with a terrain agent to run a survey operation, and the terrain agent on end2 of the terrain-to-terrain interaction directed link has status of compromised. Time to react is the amount of time surpassed from terrain agent compromise until one of the defender

agents of the cyber team reads the compromise information and adds it to the compromised terrain array variable.

3.6.1 Computation

Define T as the set of terrain agents as subset of all agents A

$$T \in A$$

Define C as the set of all compromises where compromise C_i occurs on T_j at time f_i

Define C_d as the set of all compromises that have been detected, where C_i was detected at time g_i

Therefore, average time to detect, TTD is calculated by subtracting compromise time from reaction time for all compromises and dividing by the number of successful restoral operations i

$$\overline{TTD} = \frac{\sum_{\{f_i, g_i\} \in C_d} g_i - f_i}{i}$$

3.6.2 Operational Considerations

Time to detect is a performance measure frequently referenced by subject matter experts and is specifically listed on the notional dashboard in the Defense Science Board Report calling for a performance measures dashboard [28]. The state of technology already allows for the tracking, reporting, and analysis of this measure, but would need stronger emphasis to put into operational practice. As stated previously, SIEM solutions are widely used in military and industry organizations. One of the primary purposes of SIEM systems is to alert on anomalous activity, especially compromised systems. This means there is a log, with a timestamp, of when a specific system became dysfunctional through malicious cyber activity. This is referred to as an incident. At this point an incident report is either automatically generated, or a cyber team member annotates one. The time between dysfunction and when the incident report is read and/or filed would provide the data for time to detect. This measure is something that is currently prioritized by cyber teams especially those of "cyber security center" type. Most corporations have a group of professionals that represent first line cyber defenders and are named something like "security operations center" or SOC. This team is a 24 hours/day, 365 days/year operation. The SIEMs they use are always on, and always monitoring. There is always someone on duty, or at least on call. People that have worked in this type of role will have stories about

late night and vacation/holiday work sessions due to an operational security issue. Time to detect is a real measure that is vital for these types of teams. There are many examples of literature where time to detect is explored for instance this large study of physics-based detection methods [55].

3.7 Time to Restore

Time to restore refers to the amount of time it takes for a cyber team to restore compromised systems. In the Cyber-FIT model, this means a set of defender agents are running restoral operations on a compromised terrain agent that has status of compromised. Time to restore is the amount of time surpassed from when a particular terrain agent has had status change to compromised, until that terrain agent has status changed to uncompromised.

3.7.1 Computation

Define T as the set of terrain agents as subset of all agents A

$$T \in A$$

Define C as the set of all compromises where compromise C_i occurs on T_j at time f_i

Define C_r as the set of all compromises that have been resolved where c_i was resolved at time g_i

Therefore average time to restore is calculated by subtracting compromise time from restoral time, divided by total discoveries

$$\overline{\text{TTR}} = \frac{\sum_{\{f_i, g_i\} \in C_r} g_i - f_i}{i}$$

3.7.2 Operational Considerations

Time to restore is also a performance measure frequently referenced by subject matter experts and is specifically listed on the notional dashboard in the Defense Science Board Report calling for a performance measures dashboard [28]. The state of technology already allows for the tracking, reporting, and analysis of this measure if organizations decide to. Like time to react, SIEM logging data can be used to quantify this measure for a cyber team. This measure would be simpler to compute than time to react because no human induced lag would be introduced. That is, the SIEM would detail the exact timestamps when the system went down, and was subsequently restored. Most cyber

security centers use visual aids where systems that are down are clearly displayed. Time to restore is visually apparent through these types of SIEM visual management systems. Carrying from the time to react measure, in a real-world environment a security center type of team would likely be more tuned to time to react, while the team is assigned to fixing the problem (incident response team) is more focused on time to restore. Put simply, one sub-team is reacting while another sub-team is restoring. We see this exact attempt at team segmentation in how the original U.S. military cyber teams were constructed in doctrine. Each team was originally made up of 39 team members segmented into five different squads: mission protect, cyber readiness, cyber support, discovery and counter-infiltration, and cyber threat emulation [56]. Over time the makeup of military cyber teams has evolved but the concept of sub-segmentation remains.

3.8 Time to Survey

Time to survey refers to the amount of time it takes for a cyber team to complete a survey mission where they need to gain a full understanding of the architecture, dependencies, and vulnerability level of a specified network(s) of computer systems related to an operational function. This measure is modeled after the recent utilization of military cyber teams being tasked with "survey missions". While not all the same, typically, this means a cyber team is tasked to provide cyber security status to various commanders and stakeholders as it relates to operational interests. There are many types of missions cyber teams are tasked with, all of which could potentially be modeled in various forms using Cyber-FIT. Some measures, like this one, are considered "mission dependent measures", that is performance is partially defined by the mission. In the Cyber-FIT model, this means mission parameters are loaded and then terrain agents generate vulnerabilities over a pre-specified amount of time. Upon completion of that time, a team(s) of defender agents run survey operations until the mission parameters have been satisfied. Time to survey is the amount of time surpassed from the time the team of defender agents began survey operations until the survey mission parameters have been achieved.

3.8.1 Computation

Define T as the set of terrain agents as subset of all agents A

$$T \in A$$

Define mission terrain, MT as the set of terrain agents assigned to the cyber team as subset of T

$$MT \in T$$

Define surveyed terrain, ST as the set of all terrain agents that the cyber team has surveyed as a subset of MT

$$ST \in MT$$

Define time survey mission began t_x, where

$$ST = null$$

Define time survey mission complete t_y, when

$$ST = MT$$

Therefore time to survey t_{survey} is computed by

$$t_{survey} = t_y - t_x$$

3.8.2 Operational Considerations

Time to survey is a mission dependent performance measure and more applicable to military cyber teams at this point. Military organizations frequently task cyber protection teams (CPT) to conduct "survey missions" [57]. Generally speaking, in the scope of the mission, the team will be expected to deploy system security tools and sensors into a specified network of cyber terrain responsible for DoD objectives. The cyber team will use their tools to scan the network systems and then create an assessment of the architectural, network, and host based vulnerability level and overall security posture. Missions like this are ideal from a team performance measure perspective because there are specific objectives to meet and time frames to operate within. Cyber teams tasked with a survey mission will normally plan the mission, execute the mission, and then provide a report. Therefore, in current operational environments, leadership already knows: how long the mission took, and how thorough the report is. In Cyber-FIT, the time to survey parameter is the time that the simulated team reports all required assets have been assessed. Further real world considerations can be expanded upon as cyber teams continue to evolve. For instance, time to survey could be considered based on whether the team is onsite versus offsite for the operation. The number of personnel needed and tools available could also help define time to complete survey mission performance measures. Although this measure is simplistic from a mathematical standpoint, its impact is substantial on military cyber resources. This measure is also impactful in the cyber security industry at large. Cyber teams worldwide are surveying systems nearly continuously with the SIEMs

attached to their networks. Organization managers would be greatly informed by understanding how much time and funding is needed to complete effective surveys. If a cyber incident occurs, and causes damage, how effective was the survey operations that have been ongoing? This type of analysis will inform resource allocation and risk management decisions.

3.9 Time to Secure

Time to secure refers to the amount of time it takes for a cyber team to complete a secure mission where they need to reduce the overall vulnerability level of a specified network(s) of computer systems related to an operational function. In the Cyber-FIT model, this means that mission parameters are loaded and then terrain agents generate vulnerabilities over a pre-specified amount of time, simulating time where the cyber team is not assigned to that terrain. Upon completion of the that time, a team(s) of defender agents run survey operations, building a list of vulnerabilities per terrain agent, and then secure operations, removing the vulnerabilities. Time to secure is the amount of time that has surpassed from the time the team of defender agents began survey and secure operations until the overall terrain vulnerability rate has been reduced to a mission defined value.

3.9.1 Computation

Recall previously defined Terrain Vulnerability Rate, TVR

Define m as the mission defined acceptable TVR

Define t_x as time secure mission began

Define t_y as time when $TVR = m$

Therefore time to secure is computed by

$$t_{secure} = t_y - t_x$$

3.9.2 Operational Considerations

Time to secure in an operational perspective is very similar to time to survey. Being a mission dependent measure, the same considerations carry over from the time to survey section, except that the parameters of success would be more difficult to define for time to secure. Identifying systems that must be assessed is considerably simpler than defining how to specifically "secure" the systems. From a host view, taking one computer system

at a time, one could define secure as either free from vulnerabilities, or near to free. But at a network level, security is much more complicated and difficult to define. A cyber team might find that the placement of certain devices has caused the network to have a routing type vulnerability that would be expensive to change. So tradeoffs are assessed such as an expensive architecture change versus an added layer of security to overcome the vulnerability. This means that human interpretation will come into play more in defining time to secure success parameters. This is further complicated by the fact that most networks are in place and have been for some time, sometimes a very long time. So, a cyber team securing a network is almost always having to contend with decisions about how to handle legacy infrastructure, in other words someone else's design decisions that are affecting the current security posture for the organization. At the time of this writing this measure is operational in the sense that cyber teams are actively conducting these types of missions and reporting back then security posture has been achieved.

3.10 Cyber Situation Awareness

One of the most widely used definition of situation awareness was developed by Endsley [58] describing it as the "perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future". This means that the situation awareness is context dependent, and individualized. Taking this concept to the team level, an aggregate of the individual's awareness would need to be contextualized. This is difficult because different individuals play different roles and therefore have different needs in terms of what they need specific to the cyber domain [59]. As definitions can vary so can computations of cyber situation awareness. For this version of Cyber-FIT the three basic parts of Endsley's original definition of situation awareness will be used, which can be broken down as 1) knowledge of current state, 2) comprehension of that state, and 3) projection of that state. This allows for calculations to occur in each tick of the simulation on each of those three parts. Since defender agent's primary goal in their operations is to decrease the vulnerability level of the terrain agents, vulnerabilities will be the mechanism to score cyber situation awareness for a team performance measure. That is, each agent's knowledge of vulnerabilities present, their comprehension of the vulnerabilities, and their projection of which operation to select next.

3.10.1 Computation

Define cyber situation awareness CSA, as a function of knowledge K, comprehension C, and projection P related to vulnerabilities

$$CSA = f(K, C, P)$$

Each value: K, C, P will represent a fraction on the range $[0,1]$

Each value is weighted by a mission defined weighting factor either α, β, γ assigned to each of K, C, P where

$$\alpha + \beta + \gamma = 1$$

Therefore

$$CSA = \alpha K + \beta C + \gamma P$$

Define T as the set of all mission terrain agents as a subset of all agents A

$$T \in A$$

Cyber team knowledge level K represents the ratio of surveyed vulnerability level of T to the actual vulnerability level of T

Define V as the set of all vulnerabilities v_i that terrain agent T_J can have

$V_{T_j,i} \neq 0 \leftrightarrow T_j$ has vulnerability i

Then, total vulnerability level TVL of T is calculated by

$$TVL = \sum_{j,i} V_{T_j,i}$$

Define V_s as the set of vulnerabilities v_k that have been discovered and the cyber team has current awareness of as a subset of V

$$V_s \in V$$

Then, surveyed vulnerability level SVL of T is calculated by

$$SVL = \sum_{j,k} V_{s_{k,T_j}}$$

Therefore, K is calculated by

$$K = \frac{SVL}{TVL}$$

Next, comprehension, C is defined. Comprehension represents a defender agent's understanding and implications of the combined vulnerability status of terrain agents they have surveyed.

Define D as the set of defender agents on the cyber team as a subset of all agents A,

$$D \in A$$

Where each D_l has an agent-level comprehension H_m described in the defender agent methods section. Cyber team comprehension is the average of the defender agent H_m

$$C = \frac{\sum_m H_m}{m}$$

Next, projection, P is defined. In the Cyber-FIT model, a defender agent is either conducting an operation, or not, which represents confusion or uncertainty. So, the ratio of agents not doing anything versus those actively engaged in operations is a proxy for their projection at any given time in the simulation.

Each D_l has operational variable o set to the following value

$$o = \begin{cases} 0, & operating \\ 1, & not\ operating \end{cases}$$

Therefore, P is calculated by

$$P = \frac{\sum_l o_{d_l}}{l}$$

3.10.2 Operational Considerations

Cyber situation awareness is the most abstract of the performance measures and would only be considered theoretical at this time. The basic structure of combining the three elements that Endsley defined (knowledge, comprehension, and projection) are meant to act as a building block that can be altered by researchers in order to experiment with

different ways to compute the K, C, and P values. Also, the weighting factors could be vastly different based on mission needs, or leadership mandates. In real world environments, applications are already being developed to gauge all three measures. Impromptu checks from software can be built into systems that simultaneously gauge awareness and guide the person through recommended steps [60], the data of which can be aggregated as a knowledge measure. U.S. Army researchers [61] have built an experimental tool to track user key strokes and general operating activity data that can be fed to machine learning applications and classify expert activities, which can be used to quantify both comprehension level and projection as team members move from one operation to another.

The purpose of situation awareness measures is not to gain a perfect understanding (it's impossible to quantify exactly how every team member is functioning, cognitively) but instead a general understanding. However, this measure is far more likely to apply to the emerging discipline of human-machine teaming. While we can't computationally measure the human brain to determine why the team member thinks in a certain way about cyber terrain vulnerabilities, we can for machines. As an example, a bot operating on a network to alert the cyber team via email, when a certain flag value changes, is computationally defined. That bot example actually defines knowledge (what data to look for), comprehension (what value represents an anomaly), and projection (send email to alert others because we need to take action).

3.11 Operational Efficiency

Operational efficiency refers to how well the team performs its operations in terms of resource utilization, and not wasting time. Generally speaking this means moving from task to task quickly and completing tasks quickly. Efficiency can be difficult to define because it's often difficult to prescribe specifically, in mathematical terms, what the ultimate output of the team is. Further, once the outputs are defined, it's very difficult to identify and then measure the input variables that may moderate the output. Sivasubramaniam et al reviewed efficiency measurement literature and analyzed which variables had the most effect on efficiency ratings in new product development environments. This research was able to identify nine distinct yet common independent variables that effect team efficiency. This type of analysis tracks closest to a cyber team efficiency measure due to the "input, process, output" (IPO) nature of the work environment [62]. In the Cyber-FIT model, tracking the timing of defender agent operation completions will be the mechanism to measure efficiency. That is, when each defender agent selects a new operation, that operation has both a severity level and time to complete requirement. The severity level is based on the mission assurance category (MAC) levels regularly used by Department of Defense leadership as a way to prioritize system acquisition and protection [63]. The higher the mission assurance category, the higher the

importance of that particular system. MAC levels are one, two, and three, and so those three values will be used in the computation of operational efficiency.

3.11.1 Computation

Define operational efficiency, OE as a function of operational time parameter, completion time, and operation severity.

Define D as the set of defender agents on the cyber team as a subset of all agents A

$$D \in A$$

Each D_i has attempted a set of operations, O where each O_j has a severity s equal to the mission assurance category of the operation, time to complete requirement parameter p, and time completed c.

Therefore, each d_i has an efficiency rating, e_i where

$$e = \sum_{o_{j,s=1}} \frac{p}{c} + 2\left(\sum_{o_{j,s=2}} \frac{p}{c}\right) + 3\left(\sum_{o_{j,s=3}} \frac{p}{c}\right)$$

As shown, the severity level of the operation weights the component calculation of the efficiency rating between one and three. Finally, team level operational efficiency would be the average of all individual agent efficiencies expressed as

$$OE = \frac{\sum_i e}{i}$$

3.11.2 Operational Considerations

Efficiency is a measure most cyber professionals can intuitively sense amongst their peers. Most teams in technology fields can recognize who the high performers are. This is clear by discussing team member performance where in most circumstances, it is well known who "gets stuff done". But there aren't any measures or metrics being actively tracked around efficiency in the field which means this measure is still theoretical at this time. In nearly any informational workplace like software development and cyber operations, those that are efficient are sought after by managers to work on teams where they have a vested interest. Moving from task to task without wasting time or becoming distracted is a key skill for cyber productivity. There is also an art to being able to

troubleshoot individually, without interrupting other team members. The aggregate of all of these decisions, skills, and abilities interact and manifest within cyber operational behavior. The current Cyber-FIT model is counting operational timing and severity, partially because it is modeling what is possible to measure in current real-world environments.

Most cyber teams will use some type of task management and/or incident response tracking system. These systems typically show who has taken ownership of a task (self-selecting or management assigned) and how long it took that individual to complete the task. Also, as activities occur related to the task, the system is updated with timestamps. For example, an individual is assigned a task to investigate a faulty system. The individual might upload a memory dump, then make comments about the incident, then upload an assessment report, then troubleshoot, and then restore the system. Each of these actions is saved and a picture of the incident from task assignment to resolution can be made clearly visible. This means that over time, a trend analysis can be completed to learn how well different individuals do on different types of tasks, what task categories are the most difficult, etc. Is the team getting faster? Are less team members needed per task? Is the team becoming more efficient?

3.12 Cyber Mission Capability Rate

Cyber mission capability rate represents how functional the cyberspace systems are to kinetic mission forces that depend on them. At a high level, for the purpose of describing this capability, military forces deployed to a conflict could be categorized into two groups: kinetic and cyber. The kinetic forces are conducting missions that are not cyberspace specific but depend on cyberspace to complete their mission. The cyber forces are working only on cyberspace systems in order to enable the kinetic forces. This means the primary purpose of the cyberspace terrain (computer systems) is to provide information to the kinetic forces, when requested. In the Cyber-FIT model, cyber mission capability rate is the ratio of information requests that friendly agents successfully read to the total information requests, weighted by criticality of the mission, and within an acceptable time to read parameter.

3.12.1 Computation

Define R as the set of all kinetic force information requests R_i each with time to read requirement parameter, p and criticality parameter c

Define F as the set of all kinetic force information fulfillments F_i with time to fulfill parameter t

Define cyber mission capability rate $CMCR$ calculated by

$$CMCR = \frac{F}{R}$$

Then, each F_i is computed according to the following function:

$$F_i = \begin{cases} 1 * c, & t \leq p \\ \frac{3p - t}{2p} * \frac{1}{c}, & p < t < 3p \\ 0, & t \geq 3p \end{cases}$$

Each R_i is set to the criticality parameter c, so

$$R_i = c$$

Then find total F and R values by

$$F = \sum_i F$$

$$R = \sum_i R$$

3.12.2 Operational Considerations

This measure is another that could become operational with that addition of sensors in the computing systems. A long-standing measure of the readiness of the U.S. Air Force is the aircraft mission capability rate (MCR). This measure is tracked by all flying units and reported to Congress periodically for review, where the Government Accountability Office prepares reports associated with MCR [64]. Aircraft mission capability rate, generally speaking, is a measure of the percentage of time an aircraft is available to fly missions. So, if an aircraft is damaged, or not available due to a safety mishap, then it is not available to perform a kinetic mission, operated by aircraft mission personnel. This correlates perfectly to the concept of cyber mission capability rate (CMCR). Just like the personnel responsible for making the aircraft available to the flight crew, the purpose of a cyber team is to make cyber terrain information systems available to kinetic mission forces. The performance of that cyber terrain in fulfilling information requests is the primary measure kinetic forces will judge the cyber terrain they depend on. This cyber team performance measure is incredibly difficult to quantitatively measure in real world

operations and would be extremely difficult to actually implement. Most users in any military or industry setting of corporate IT systems have an intuitive sense of how the computer systems are working based on responsiveness they are experiencing when using and accessing computer systems. In that sense, a survey could be sent out periodically to get a qualitative assessment of cyber mission capability rate.

A quantitative measure would require increasing the sensing capacity within the computer network. For instance, servers sending information through a firewall would have to log and aggregate response packets by mission identification. In real world operations, there are typically many different cyber teams working on different aspects of different missions. Disentangling the various operations and cyber team protective behaviors is the primary difficulty with operationalizing this measure. Also, the actual desired mission capability rate would be difficult to define and depend on how it is measured. In obvious and simple terms, the higher the rate the better. But the way a team in one location defines rate would likely differ from another team somewhere else. One team might define responses fulfilled as all the same while another might use speed of information as in indicator (like this section defines rate). For aircraft mission capability, the definition is clear, simple, and it can be applied to all aircraft types. The amount of available flying hours is straightforward and easy to understand. If implemented, cyber mission capability rate should seek to define it in a way like aircraft that could be applied across different types of computer systems, and easily measured in a consistent way. Where the mission capability rate is likely most similar to aircraft mission capability rate would be the range of values. Older legacy systems such as SCADA would likely be lower whereas new cyber terrain supporting F-35 flight tests would have a higher mission capability rate.

3.13 Time to Compromise

Time to compromise represents the amount of time it takes from when the attacker starts an attack campaign until targeted machines are compromised. In the Cyber-FIT model, this is measured from the time an attacker begins phase one of an attack campaign until, during the exploitation phase, a terrain agent changes state from operating to compromised.

3.13.1 Computation

Define O as the set of attacker agents on the cyber team as a subset of all agents A

$$O \in A$$

Each O_i has a total attack campaign time parameter t and number of successful attacks s

Define time to compromise TC for each O_i computed by

$$TC = \frac{t}{s}$$

3.13.2 Operational Considerations

This cyber team performance measure is attacker based and theoretical at this time. This is largely due to the fact that the details of the attacker's activities must be available in order to compute it. Agent-based modeling and simulation software can provide an excellent mechanism to experiment with phenomenon like this. Obviously, it would be of great interest to military and industrial organizations alike to have a full understanding of when, where, and how cyber adversaries begin attack campaigns, and when they become successful and on what systems. The Cyber-FIT model, and agent-based systems in general, can be an excellent tool to try out ideas on what might be possible, in a computational and programmatic manner. Research in this area is available such as a modeling framework for detecting and assessing the impact of different attack types [65]. Running simulations can lead to theories about what exactly is going on with real world systems. Then, the empirical data an organization actually has can be compared to simulated data. This can either validate, at some level, the simulation software, or give clues as to why the simulation is not outputting data that matches empirical data.

Another difficulty with this measure is attribution. If an organization finds that several different systems are compromised, there's no way to immediately know that these are related. Was this one attack with several victims? Was this several different activities from different malicious actors? The most realistic application of this measure would be from the attacker's view. An attacker would know when precisely they start an attack and when they realize the goal. From an organizational defense perspective, this is the purpose of "red-teaming" and cyber threat emulation. The organization can practice simulated attacks against itself to discover problems, which can be measured (time to attack) [66].

3.14 Compromise Success Rate

Compromise success rate represents how successful attackers are in an attack campaign. It is measured by number of successful attacks and number of attack attempts. In the Cyber-FIT model, this is measured by continuously counting, each tick, how many total attacks have been attempted by each attacker agent and how many of those attacks have been successful.

3.14.1 Computation

Define O as the set of attacker agents on the cyber team as a subset of all agents A

$$O \in A$$

Each O_i has a total number of attacks x during a campaign, and number of successful attacks s

Define compromise success rate CSR for each O_i computed by

$$CSR = \frac{s}{x}$$

3.14.2 Operational Considerations

Much like the previous measure, this is also attacker based and theoretical for the same concerns. In real world systems, knowledge about how many attacks have been attempted is very difficult to quantify. In situations where an attack has been successful, in nearly all cases, there is very little in the way of how many other attacks the adversary launched that weren't successful. The concept of "covering your tracks" means attackers tend to be as careful as possible about not giving away their position and removing evidence as they go. This is detailed as an exploitation technique by the MITRE tracking system [67]. Also, in most circumstances, the cyber teams and organizational leadership do not initiate forensic investigation of attacks until well after the attacks have been executed. Going back in time through logs and SIEM data is extremely time consuming and resource intensive. Like time to compromise performance measure, compromise success rate is virtually unknown to the organization on the receiving end of attacks.

3.15 Force-Force Interaction Network Node Total Degree Centrality

Force-force interaction network node centrality total degree is a measure meant to detect key leaders within the cyber team. This is done by examining a dynamic network of communications within the cyber team where each team member is a node. In the Cyber-FIT model this is done by creating directed links from defender agent to other defender agents in order to share vulnerability and compromise information. At every tick, some number of defender agents may communicate with others, in which case a directed link between them forms for a random time period in order to communicate. Throughout the simulation, Cyber-FIT stores this date as a file of links. Post-simulation processing software converts the link data to a time period based dynamic network file that is imported into ORA [68] for dynamic network analysis. ORA processes the data, runs network

science algorithms on it, and provides a report detailing selected network measures, in this case node total degree centrality.

3.15.1 Computation

Define A as the input network with n nodes (each representing a defender agent's ego) and maximum link value v, representing the number of messages sent to other defender agents

Total-degree centrality for each defender agent node i, TDC calculated by

$$TDC_i = \frac{\sum_j (A_{i,j} + A_{j,i}) - A_{i,i}}{2v(n-1)}$$

3.15.2 Operational Considerations

This measure is one of two network science based cyber team performance measures, along with terrain-terrain interaction network density. Both are considered theoretical (and proof of concept) at this time. As a cyber team works together, communication networks emerge and dynamically change over time. Capturing the network data in periods of time (minutes, hours, days, etc.) and then analyzing how the measures change over time is called dynamic network analysis. Dynamic network analysis has been used to cover a wide array of scientific questions, especially those in the social sciences where human interactions are the core data being considered [69]. There are many network measures that could be considered, in order to gain insights about how a team is performing, so in this version of Cyber-FIT, the measures were limited to two. This is so the efficacy of network science measures could be considered in a simulation system with two of the most frequently used measures. This measure, node total-degree centrality, is a popular measure used in many studies to identify nodes most important for the flow of information [70]. This relates directly to cyber team performance because in many cases, the key leaders of an operation are not readily apparent based on the formal organizational structure. In an operational environment very similar data to the Cyber-FIT simulation data could be extracted and analyzed. The easiest way to do this would be to export the chat data from a team messaging server, especially when a cyber operation is being conducted by team members not physically in the same space.

3.16 Terrain-Terrain Interaction Network Density

Terrain-terrain interaction network density represents how much of the computer network is connected at any given time. In the Cyber-FIT model, this is simulated when directed links are created between terrain agents as a result of defender, friendly, or attacker agent behavior. Each time, a defender, friendly, or attacker agent creates a directed link to

a terrain agent (which is a force-terrain interaction), one or more subsequent directed links are created between that terrain agent and other terrain agents. Then, at any given tick, a network of directed links where each end is of type terrain agent, can be extracted from the simulation.

3.16.1 Computation

Define A as the binary input network of terrain-terrain directed links with m rows and n columns

Density D is computed by

$$D = \frac{\sum(A)}{m \times n}$$

3.16.2 Operational Considerations

This measure is the other network science type measure included in this version of Cyber-FIT. Networks will usually have similar traffic patterns over time, based on patterns of life and usage by human involvement. This is why network density shows indications of a potentially effective measure to use for computer network visualization and monitoring [71]. A corporate network with normal business hours will have very different traffic patterns at 2:00 PM versus 2:00 AM. Network density as a cyber team performance measure is less an indicator of performance, and more a corollary to overall mission metrics. That is, there will not be a specific network density measure the team is aiming for. Instead, network density can be used as an indicator of normal operations, versus adversarial anomaly detection.

4 Chapter 4: Cyber-FIT version 4

At the lowest level, Cyber-FIT is made up of agents and interactions. All agents are one of two main types: forces and terrain. From a military modeling and simulation perspective this is the highest-level categorization of agent types and allows for future model output porting and multi-modeling. Force agents represent military personnel in a conflict simulation and has three sub-types: defender, attacker, and friendly. Terrain agents represent the computer systems present in a cyber conflict simulation and has three sub-types: networking, serving, and host. Terrain agents, representing computers, are named as such due to the United States Department of Defense creating US Cyber Command and declaring cyberspace a terrain of war [72]. The interactions between agents are either force-to-force, force-to-terrain, or terrain-to-terrain.

4.1 Terrain Agents

The terrain agents represent cyber terrain: any computing machine that military forces depend on. This can include servers in a data center, a tablet used in field operations, laptops in a work center, etc. Terrain agents are the cyberspace assets that military cyber forces are vying to control. In this version, terrain agents are all owned by the defender agent side of the conflict. This simulates a deployment of a cyber team and focuses the development and computational modeling on performance measures defining success for that deployment. As designed in the current version of Cyber-FIT, terrain agents are one of three type: networking, server, or host. This is based on the typical three tier architecture that most every corporate environment deploys to rout and serve information to host machines used by individuals. This doesn't preclude additional terrain types in future versions of Cyber-FIT, in fact additional agent types are expected to be modeled for more specific use cases. For example, the next candidate might be mobile terrain agents. This would mean that a new type would be added in the agent and rulesets would change affecting how other agents react in the event an interaction takes place. Modeling a mobile device would likely consider how signal processing must be incorporated [73]. Also, the differences in how attacks manifest in mobile systems must be considered [74]. Terrain agent class behaviors and variables are defined in the following table.

Variables	
Name	Description
Type	Type of cyber terrain, either networking, server, or host
Status	Either operating normal, or compromised
vulnerabilities[]	List of vulnerability identification numbers that are

	currently present
payloads[]	List of payloads delivered by attacker agent currently present
missionID	Kinetic mission identification number this terrain agent is supporting
Behaviors	
Name	Description
step()	Every step a terrain agent will generate a random number of vulnerabilities that are now present, update its own terrain statistics, and then set its color for simulation visualization
generateVulnerabilities()	Each tick, a vulnerability might occur. Vulnerabilities are denoted by a vulnerability number between 0 and 99 that represents the severity of the vulnerability. The higher the number, the more severe the vulnerability, except for zero which represents a zero day vulnerability that can only be exploited by the most sophisticated adversaries
updateTerrainStats()	Update agent's own statistics
createConnection()	Connects to another terrain agent for computing purposes
addZeroDay()	Adds zero day vulnerability to itself due to attacker agent successfully developing and delivering a zero day vulnerability to it
sendMessage()	Connects to another terrain agent in order to send information message for defender or friendly agents
trySurvey()	Tries a survey operation initiated by a defender agent, which results in either a success, where terrain agent info is passed back to defender agent, or a fail, where the survey was not successful and no information was passed back to the defender agent
trySecure()	Tries a secure operation initiated by a defender agent, which results in either a success, where vulnerabilities identified by the defender agent have been removed, or

	a fail, where the identified vulnerabilities have not been removed
tryRestore()	Is in a compromised state, and tries a restoral operation initiated by a defender agent, which results in either a success, where the compromised terrain agent is restored to working or fail where the terrain agent is still compromised
tryAttack()	Is in an uncompromised state, and tries an attack where a payload that has been delivered by an attacker agent is either successful due to existing vulnerabilities, where the terrain agent becomes compromised, or a fail where the terrain agent continues working normally

Table 15: Terrain Agent Class Variables and Behavior Methods

4.2 Defender Agents

The defender agents represent the cyber forces deployed to the conflict with a mission to defend the cyber terrain that kinetic forces depend on to carry out their own missions. Defender agents are deployed to the simulated conflict as teams of any size, made up of members of any type, as denoted by the cyber forces configuration file. Once deployed, the defender agents will work together to share information about the cyber terrain, remove vulnerabilities from assigned terrain, and restore terrain that are compromised. All of their behaviors are based on some subset of their class variables, depending on the circumstances of the current run. Defender agent class behaviors and variables are defined in the following table.

Variables	
Name	Description
Team	Cyber team identification number
Squad	Represents the sub-team that the defender agent is assigned. There are three squad types: lead, network, and host. Lead represents the team leadership and intelligence operations. Network defender agents focus on vulnerable terrain that are networking and serving type. Host defender agents focus on vulnerable terrain that are serving and host type
knowledge	Knowledge level denoted as low, medium, or high

Skill	Skill level denoted as low, medium, or high
experience	Experience level denoted as low, medium, or high
compromisedTerrain[]	List of terrain the defender agent believes to be compromised
vulnerabilitiesTerrain[:]	Table of terrain agents and vulnerabilities each terrain agent that the defender agent believes exist on that terrain
opType	Type of cyber operation currently working on
opTime	Current time working on current cyber operation
totalOps	Total number of cyber operations conducted
totalSurveySucces	Total number of successful survey attempts
totalSecureSucess	Total number of successful secure attempts
totalRestoralSuccess	Total number of successful restoral attempts
Behaviors	
Name	Description
step()	Every step a defender agent will either: continue restoral operations on compromised terrain, continue the current cyber operation they were working, or select a new cyber operation to begin
getNewOp()	Process defender agent goes through to select a new cyber operation to begin next step. The operations that the defender agent can select are one of seven types as defined CISA [75]: Analyze, Collect and Operate, Investigate, Operate and Maintain, Oversee and Govern, Protect and Defend, and Securely Provision.
continueOp()	Defender agent has not completed current cyber operation so it continues that cyber operation
interactWithForce()	Defender agent, based on their current cyber operation, needs to interact with another defender agent for communication purposes
interactWithTerrain()	Defender agent, based on their current cyber operation, needs to interact with terrain agents. This represents the cyber operations where a defender agent is

	attempting to survey, restore, secure, or message
surveyOp()	Defender agent, based on their current cyber operation, needs to use cyber terrain to survey other cyber terrain in order to update their cyber situation awareness
secureOp()	Defender agent, based on their current cyber operation, uses cyber terrain to connect to other cyber terrain in order to increase the cyber security of those terrain agents by removing vulnerabilities
restoralOp()	Defender agent is aware of compromised terrain and has been assigned to task of attempting to restore that terrain
hasCompromiseSA	Defender agent has become aware, or still is aware of compromised terrain agents that are compromised. Defender agent will share that information with other members of the cyber team
sendMessage()	Sends message to teammate based on current operation
sendMessageCompromised()	Has information about compromised terrain so shares that information with teammates or team lead

Table 16: Defender Agent Class Variables and Behavior Methods

4.3 Attacker Agents

The attacker agents represent the cyber forces assigned to attack the cyber terrain that the defending and friendly forces depend on for their military operations. In cyber war-gaming and exercises this is commonly referred to as OPFOR (opposing forces). Any number of attacker agents can be added to the conflict, with each attacker agent having a sophistication level as denoted by the simulation configuration files. The attacker agents work alone. Attacker agents work through the cyber kill chain as defined by Lockheed Martin [47] with the ultimate goal of compromising terrain agents. Once compromised, friendly forces cannot utilize those terrain agents to conduct kinetic operational missions. The modeling of how an attack works is based on the MITRE ATT&CK® framework [76]. Terrain agents must be vulnerable to an attacking technique by an attacker agent. As of this writing, ATT&CK has 215 techniques documented and described. In this version of Cyber-FIT, attacker agents have 100 techniques available. In real world operations, the 215 techniques could each exploit one to many vulnerabilities present on a computer network. To abstract that away, in this version of Cyber-FIT, vulnerability identification numbers and attack technique numbers are representing a similarity (attack matches

vulnerability) that allows the attack to be successful. This level of complexity is by design so different taxonomies can be implemented in future versions. Attacker agent class behaviors and variables are defined in the following table.

Variables	
Name	Description
Tier	Sophistication level of the attacker agent as defined by the Department of Defense [28].
Phase	Current phase of the cyber kill chain that the attacker agent is engaged in
phaseTime	Amount of time spent in the current phase of the cyber kill chain
recons[]	List of terrain agent identification numbers that the attacker agent was able to successfully conduct cyber reconnaissance operations on
attacks[]	List of attack identification numbers that are currently available to the attacker agent
deliveredTo[]	List of terrain agent identification numbers that the attacker agent was able to deliver cyber payload to
attackAttempts	Number of terrain agents the attacker agent attempted an attack on during the current simulation
attackSuccesses	Number of terrain agents the attacker agent successfully compromised
Behaviors	
Name	Description
step()	Every step, an attacker agent continues on in whatever phase of the cyber kill chain it is in. If there has been an interruption, the attacker continues in “phase zero”, simulating time between or before beginning an attack attempt
initialize()	Attacker agent initializes number of attacks and attack identification numbers available before starting the cyber kill chain and after every cyber kill chain attempt

reconPhase()	Attacker agent attempts to connect to terrain agents and discover vulnerabilities
weaponizationPhase()	Attacker agent spends time on one terrain agent preparing attacks to be delivered to other terrain agents
deliveryPhase()	Attacker agent delivers payload to terrain agents it believes to be vulnerable to that particular attack based on information gathered during recon phase
exploitationPhase()	Attacker agent waits as exploit is attempted by malicious code on terrain agent
commandAndControlPhase()	Attacker agent is able to interact with select compromised agents for the purpose of controlling that terrain agent and furthering attack objectives
actionsOnObjectivesPhase()	Attacker agent waits as further actions occur on own terrain

Table 17: Attacker Agent Class Variables and Behaviors

4.4 Friendly Agents

The friendly agents represent the military forces conducting kinetic missions associated with the simulated conflict. In order to achieve their objectives they depend on information and computing resources provided by the cyber terrain. Therefore, at any given time, friendly agents might connect to terrain agents associated with their mission to request information. These information requests are processed and, depending on the terrain agent status, fulfilled with a timing value or not fulfilled. Information requests have a mission assurance category level between one and three based on the Department of Defense assigned criteria. Friendly agent class behaviors and variables are defined in the following table.

Variables	
Name	Description
missionID	Kinetic mission identification number this friendly agent is assigned to
infoRequests	Total number of information requests made during simulated mission
infoFulfills	Total number of information request fulfillments during a simulated mission
Behaviors	

Name	Description
step()	Every step a friendly agent may or may not connect to a terrain agent and make an information request

Table 18: Friendly Agent Class Variables and Behaviors

4.5 Force-Force Interaction Links

The force-to-force interactions are directed links representing force agents interacting within a simulated cyber conflict. The force-to-force agent interactions are informational in nature and related to either the cyber or kinetic operation currently being conducted by force agents. Force-to-force interaction link class variables and behaviors are defined in the following table.

Variables	
Name	Description
lifetime	Number of ticks this link will remain active
type	Type of link
Behaviors	
Name	Description
step()	Decrement lifetime value and if equal to zero link will die

Table 19: Force-Force Interaction Link Class Variables and Behavior Methods

4.6 Force-Terrain Interaction Links

The force-to-terrain interactions are directed links representing force agents interacting with cyber terrain agents during a simulated cyber conflict. Force-to-terrain agent interactions occur when any force agent (defender, attacker, or friendly) needs to utilize a terrain agent for any reason. Force-to-terrain interactions can occur because agents need to use terrain to message other agents, read information, update terrain, or send messages. Force-to-terrain interaction link class variables and behaviors are defined in the following table.

Variables	
Name	Description
lifetime	Number of ticks this link will remain active

type	Type of link
Behaviors	
Name	Description
step()	Decrement lifetime value and if equal to zero link will die

Table 20: Force-Force Interaction Link Class Variables and Behavior Methods

4.7 Terrain-Terrain Interaction Links

The terrain-to-terrain interactions are directed links representing terrain agents interacting with other terrain agents during a simulated cyber conflict. Terrain-to-terrain agent interactions occur when terrains are connecting to each other simulating all of the cyber operations and interactions built into this model. For example, when an attacker agent is using a cyber terrain agent to simulate the installation of malicious software onto a friendly agent's cyber terrain, an attacking type terrain-to-terrain agent interaction is created. Terrain-to-terrain interaction link class variables and behaviors are defined in the following table.

Variables	
Name	Description
lifetime	Number of ticks this link will remain active
type	Type of link
Behaviors	
Name	Description
step()	Decrement lifetime value and if equal to zero link will die

Table 21: Terrain-Terrain Interaction Link Class Variables and Behavior Methods

4.8 Cyber Team Performance Simulation

A cyber team performance simulation is conducted in order to output all performance measures for analysis and implications. The cyber team will deploy to a simulated conflict involving 500 computer systems operating to support 4 kinetic missions that need to be protected from several adversarial cyber forces of varying tiers. When initializing Cyber-FIT, three files must be configured to setup key variables along with mission information. The first file, called missions supported, defines the friendly kinetic mission cyber terrain to be defended, including number of forces and associated cyber

terrain systems. The second file, called, defenders, defines the cyber teams that will deploy in terms of squad, knowledge, skill, and experience. The third file, called attackers, defines the adversarial forces in terms of numbers and sophistication level. The following three tables display the pertinent contents of each file.

Mission ID	Unit	Friendly Forces	Networking Terrain	Server Terrain	Client Terrain
0	Base	0	10	20	30
1	Command Post	25	5	10	50
2	Fires	75	5	6	225
3	Logistics	50	3	5	75
4	Security	75	2	4	50

Table 22: Summary of Simulation Missions Supported File

Cyber Team ID	Squad	Knowledge	Skill	Experience
1	1	2	2	2
1	2	1	1	1
1	2	2	2	2
1	2	2	2	2
1	2	3	3	3
1	3	1	1	1
1	3	2	2	2
1	3	2	2	2
1	3	3	3	3

Table 23: Summary of Simulation Cyber Teams File

Adversary Type	Tier
State	3
Criminal	4
State	5

Table 24: Summary of Simulation Adversaries File

The simulation is run for 14,400 ticks, with each tick representing one simulated minute of time. This represents a ten-day simulation of continuous cyber conflict. Each run of the simulation takes approximately twenty minutes to complete on a Dell computer running Windows 10 with an Intel Xeon 2.7 GHz processor and 32 GB of RAM. Each simulation produces approximately 6 MB of team performance data. This simulation was run ten times.

4.9 Simulation Results

This section presents the sixteen cyber team performance measures resulting from the cyber conflict simulation.

4.9.1 Terrain Vulnerability Rate and Change Results

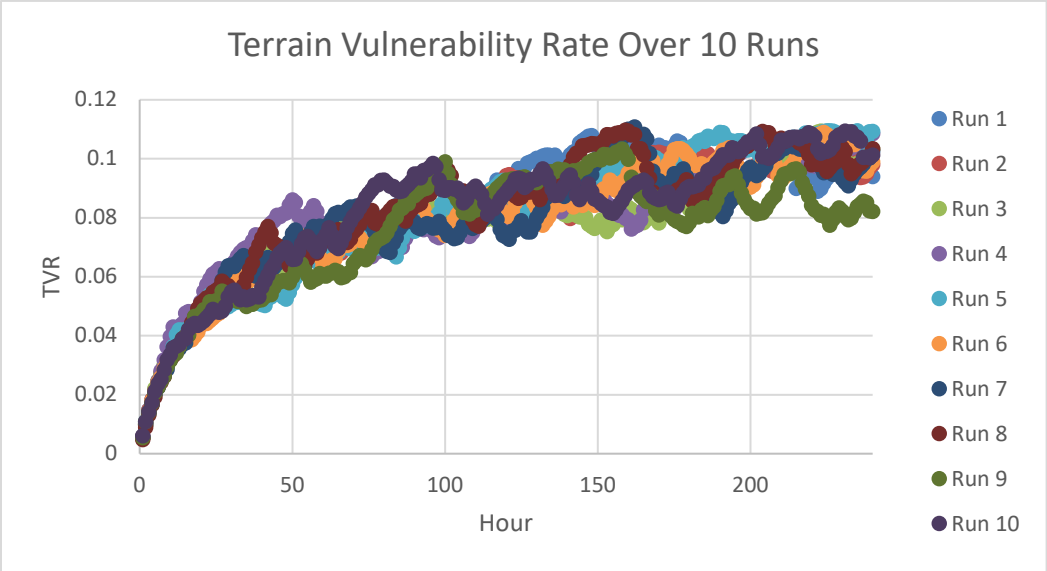


Figure 20: Terrain Vulnerability Rate

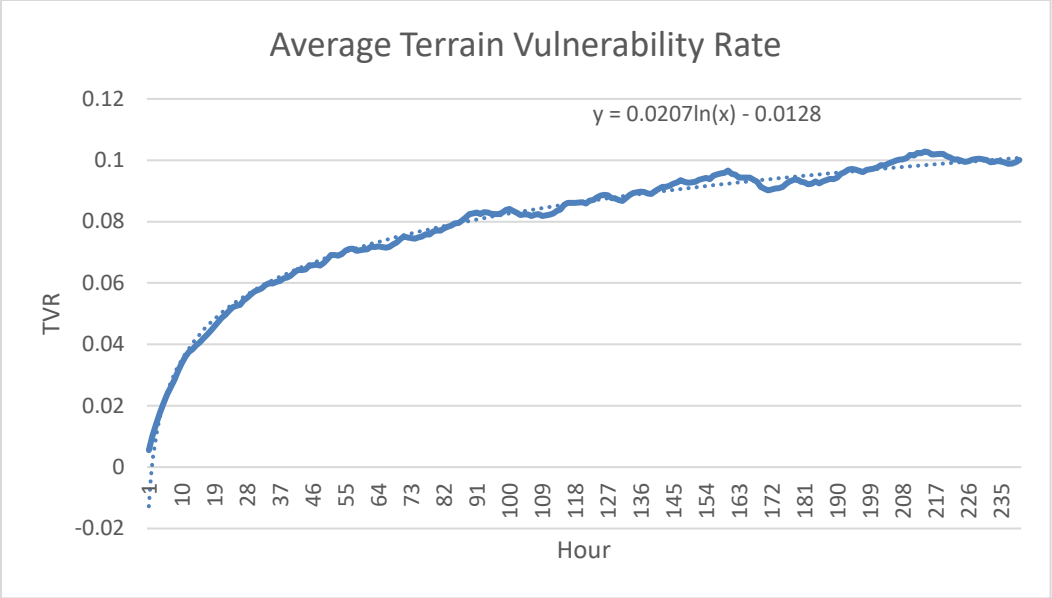


Figure 21: Average Terrain Vulnerability Rate

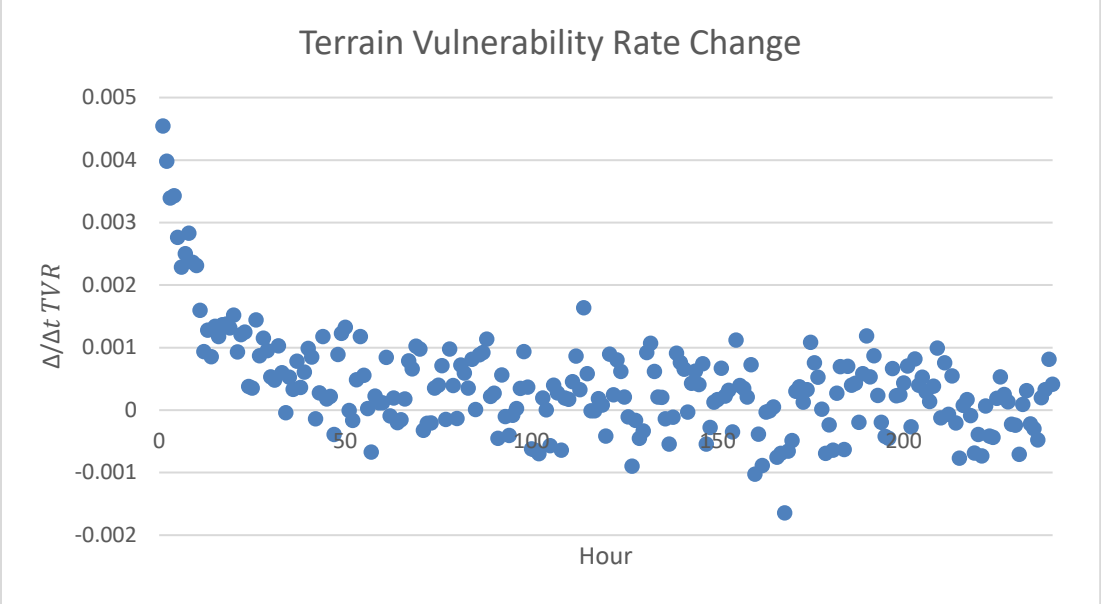


Figure 22: Average Terrain Vulnerability Rate Change

4.9.2 Terrain Vulnerability Rate and Change Discussion

As shown in Figures 20 and 21, the terrain vulnerability rate will increase rapidly in the early part of the simulation and then level off. A logarithmic best fit curve to the average terrain vulnerability rate, as shown in Figure 21, is $y = 0.0207\,ln(x) - 0.0128$, meaning the terrain vulnerability rate would not increase very much as the simulation continues under the configuration parameters. After tick 150, as shown in Figure 22, the

change value hovers around zero. From a realism perspective, this is what is expected from most normal operations: the terrain vulnerability rate staying steady. In the last hour of the simulation, the lowest terrain vulnerability rate simulated is .082 and the highest is .109. The terrain vulnerability rate value realized in this simulation is approximately 0.1, which means a ten percent vulnerable state. This number is an abstraction and not meant to match real world operations perfectly. In real world situations, ten percent vulnerable is likely too high. However, in the simulation this means each simulated computer has approximately ten vulnerabilities at any given time, out of one hundred possible vulnerabilities. In real world operations, vulnerability state is well known through the use of network vulnerability management software where computers on the network report back about known vulnerabilities. Over time trend analysis can give a sense of how well the cyber team is managing vulnerabilities and therefore their own performance.

4.9.3 Terrain Compromise Rate, Change and Time Results

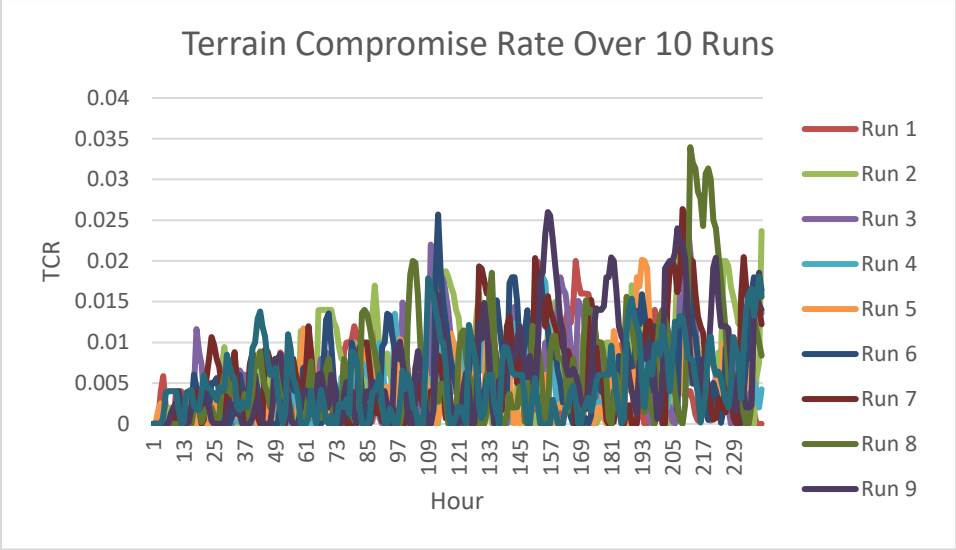


Figure 23: Terrain Compromise Rate

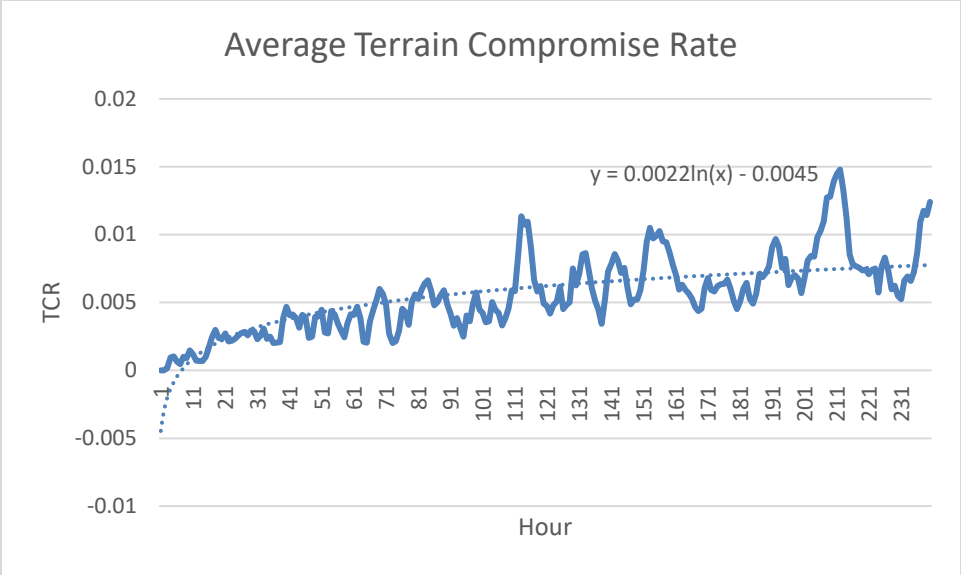


Figure 24: Average Terrain Compromise Rate

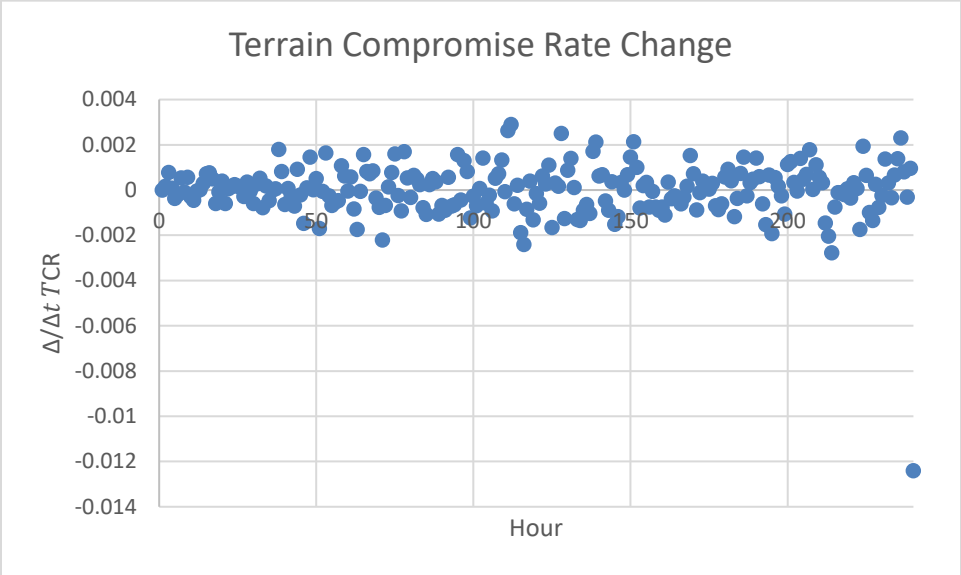


Figure 25: Terrain Compromise Rate Change

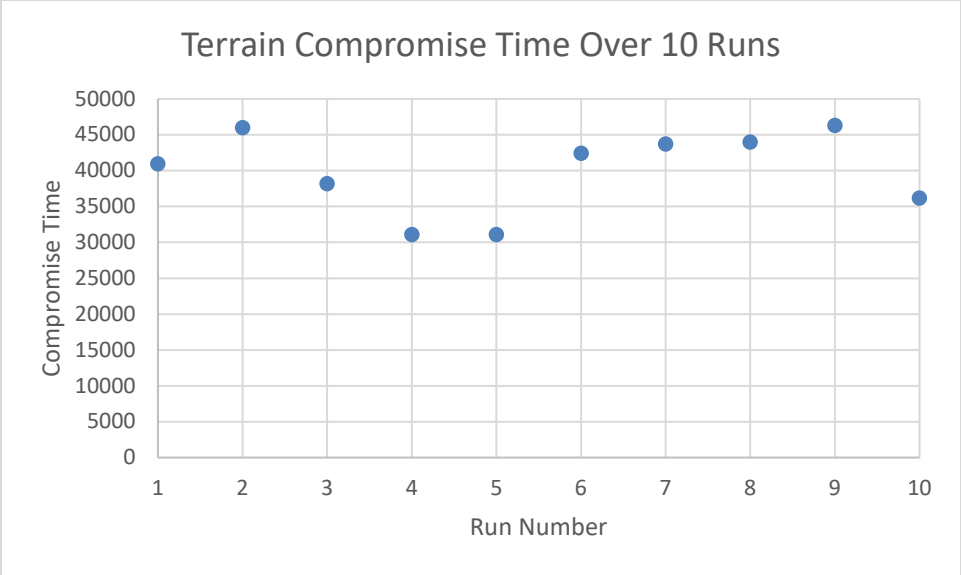


Figure 26: Terrain Compromise Time

4.9.4 Terrain Compromise Rate, Change, and Time Discussion

Terrain compromise rate, as shown in Figures 23 and 24, is more variable then terrain vulnerability rate. This is likely the case in real world operations as it is easier to detect vulnerabilities than compromised systems. Usually, a compromised system is unknown for some time, which is precisely what attackers are trying to do: compromise systems unbeknownst to the cyber team. Therefore, terrain compromise rate, time, and change will be harder to track in real world operations. The best fit logarithmic curve to average terrain compromise rate is $y = 0.0022ln(x) - 0.0045$. Like terrain vulnerability rate there is an initial increase, with the curve sloping up and then a leveling off. At that point, the attacking cyber team and defending cyber team are in a protracted battle where each side is fairly evenly matched. The attackers are able to exploit some systems, and the defenders are able to eventually identify and restore the compromised systems, which accounts for the variability in the curve. This is clearly shown in Figure 25, which is terrain compromise rate change, where the values hover around zero the entire simulation.

In real world operations, this value would be near zero, nearly all of the time. In terms of this simulation, the model was designed to ensure some attacks would be successful, or else there would be nothing to analyze. Also, one of the primary purposes of an agent-based model is to conduct what-if analysis. This means searching for the combination of parameters that does cause differences in terrain compromise rate, and other dependent variables. Also, this is a ten-day (240 hour) simulation. A more realistic way to simulate successful computer compromises would be over a much longer time horizon. Figure 26 shows total compromise time for each run of the simulation. This value

represents how long each computer (in aggregate) was off-line and inaccessible over the course of the ten days. The minimum value simulated was 31,068 minutes and the maximum was 46,315. In real world operations, this value is usually better known. That is, when a machine goes offline, and stops checking into servers, there is a log entry for when this occurs. So, total downtime for machines can be tracked fairly precisely. The more difficult part is attributing downtime to malicious activity. Some machines can go offline for completely benign reasons ranging from operating system error, user locking a computer out, hardware problems, infrastructure work, etc. Typically, downtime is monitored very closely, as most computers on a network are there for a purpose and when on, are needed to do some type of job. In the Cyber-FIT model, all downtime is related to malicious activity, in the real world this would have to be decoupled.

4.9.5 Time to Detect and Time to Restore Results

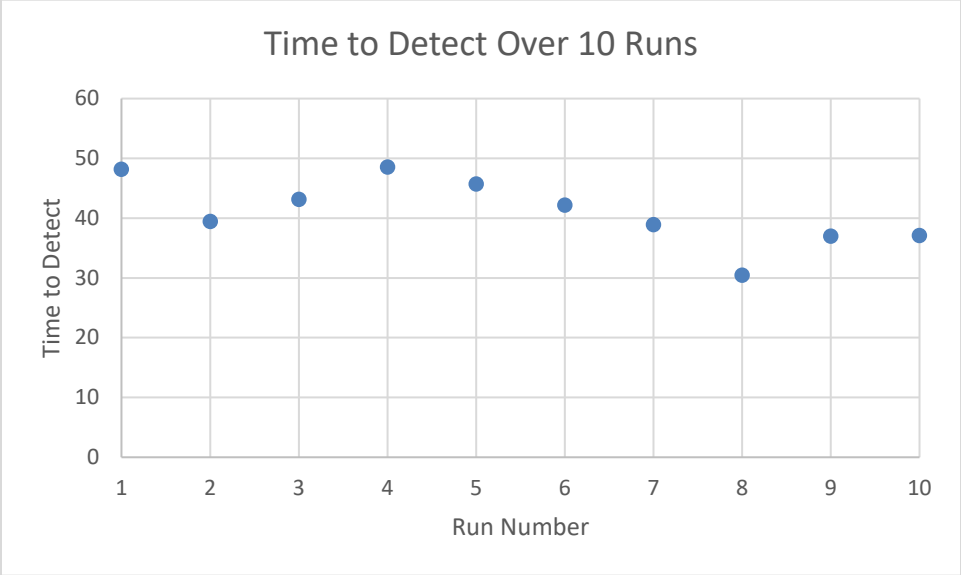


Figure 27: Time to Detect

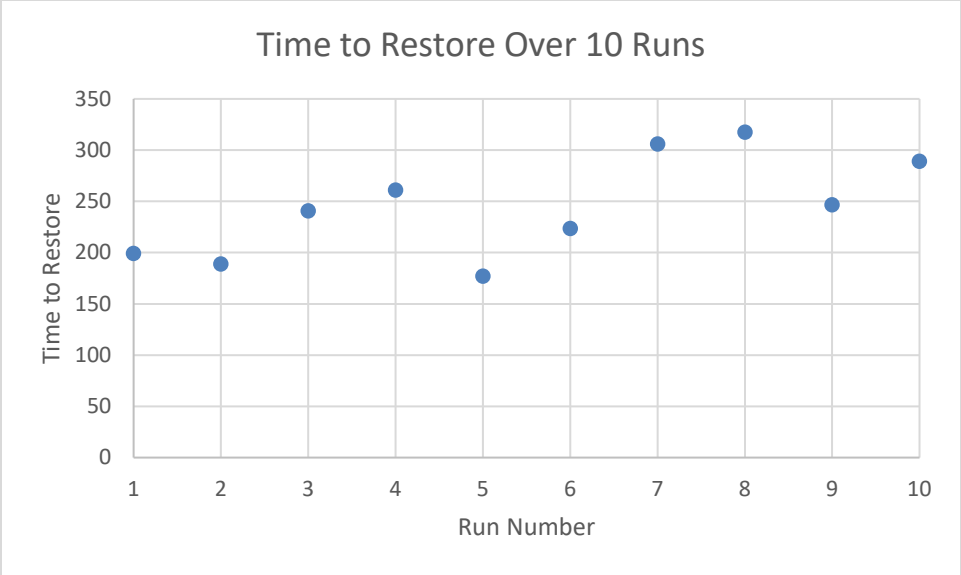


Figure 28: Time to Restore

4.9.6 Time to Detect and Time to Restore Discussion

Figure 27 shows the average time it took the cyber team to detect a machine was compromised over each run of the simulation. The minimum average time to detect was 30.44 minutes and the maximum average time to detect was 48.57. Figure 28 shows the average time it took the cyber team to restore compromised systems. The minimum time average time to restore was 177.16 minutes and maximum average time to restore was 317.56. Figure 28 shows the model results in higher variance for time to restore, which probably matches reality. In real world operations, these performance measures would both probably be high performing. Consider that the most devastating malicious compromises go unnoticed for long periods of time. This is why ransomware attacks are so popular (the cyber team cannot remove the malware) and exfiltration attacks (where large amounts of information is stolen) are so worrisome for corporations. In real world operations it is usually difficult to precisely determine how long a compromise went on unnoticed. This is largely due to the fact that it takes resources to investigate after the fact.

4.9.7 Cyber Situation Awareness Results

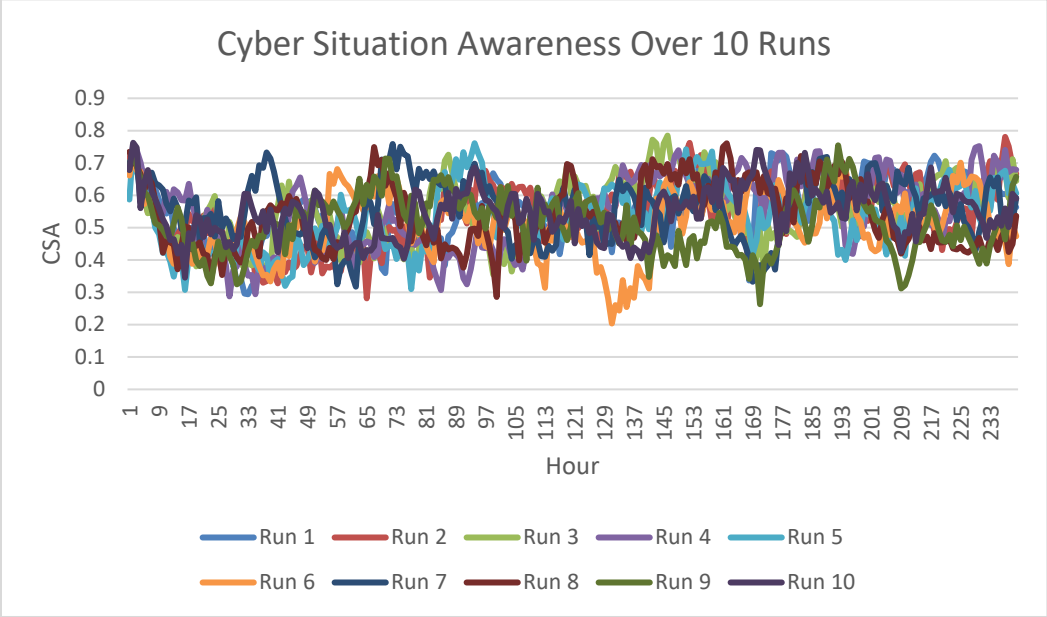


Figure 29: Cyber Situation Awareness

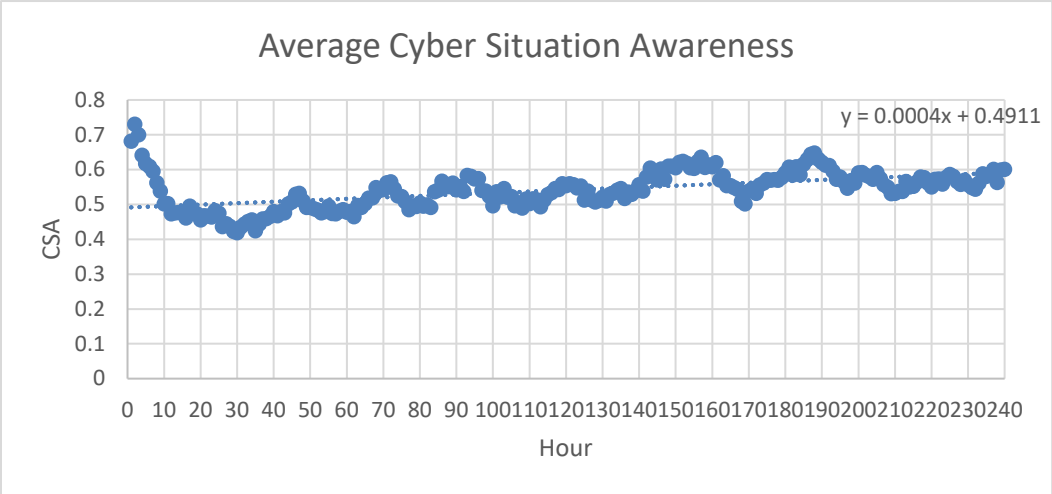


Figure 30: Average Cyber Situation Awareness

4.9.8 Cyber Situation Awareness Discussion

Figure 29 shows the model resulting in fairly high variance in cyber situation awareness over runs of the simulation. Figure 30 shows the average cyber situation awareness starting off higher and then decreasing (this is due to the computers becoming vulnerable and the cyber team needing to survey in order to find and become aware of the vulnerabilities). The values level off after the tenth hour. After this point, the minimum cyber situation awareness simulated is 0.204 and the maximum cyber situation awareness

simulated is 0.785. The best fit linear curve to the average cyber situation awareness is $y = 0.004x + 0.4911$. The positive slope means that cyber situation awareness will continue to increase over time. This likely correlates well with real world operations due to teams sharing information over time and continuously communicating. This performance measure exists at this time in a theoretical sense only. That is, there are no real world teams actively tracking cyber situation awareness. It continues to be a concept understood, and sometimes discussed, and usually perceived, but not actively monitored.

4.9.9 Cyber Mission Capability Rate Results

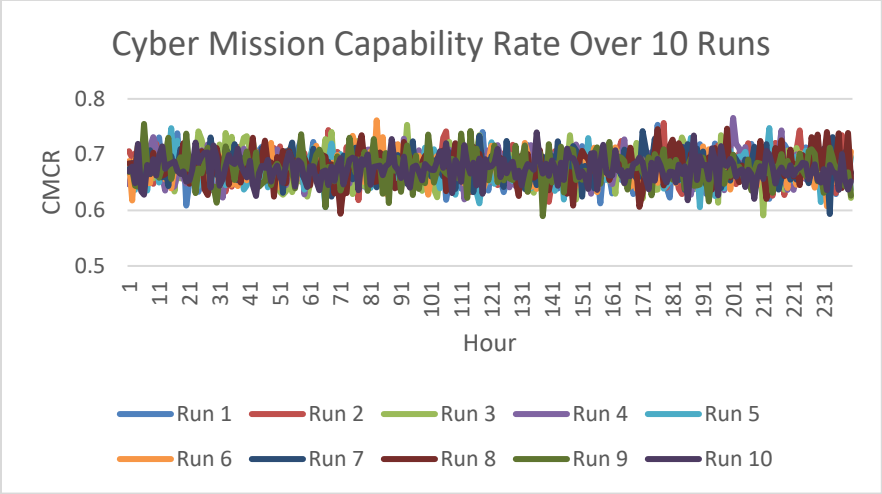


Figure 31: Cyber Mission Capability Rate

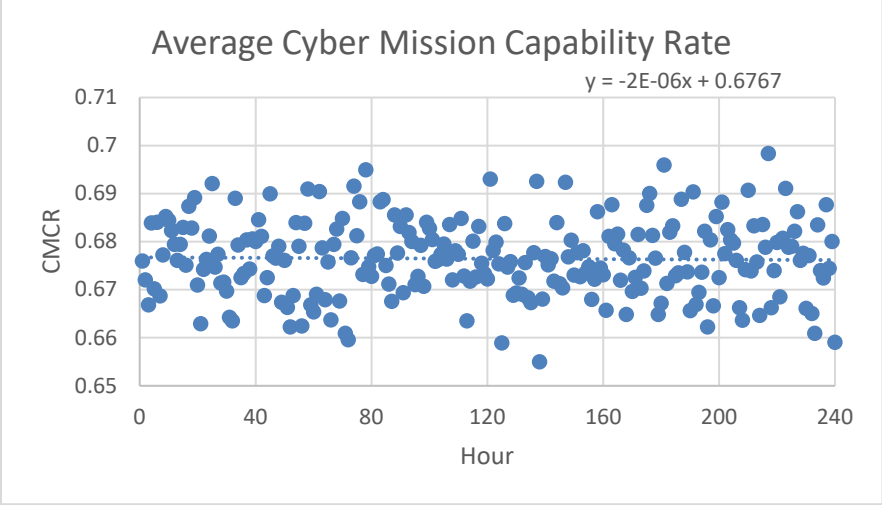


Figure 32: Average Cyber Mission Capability Rate

4.9.10 Cyber Mission Capability Rate Discussion

Figure 31 shows a highly variant but steady cyber mission capability rate over the ten runs of the simulation. Recall, cyber mission capability rate represents how well the computer network is providing the information requests needed by friendly forces to operate their own missions. This seems to be the most basic and important representation of what the purpose of the cyber team is: ensure the network moves information to those who need it. Like many cyber performance measures over time, this data is captured at hourly points throughout the ten-day simulation. The minimum cyber mission capability rate simulated was 0.589 and the maximum was 0.766. As shown in Figure 32, the slope of the best fit linear curve to the average cyber mission capability rate is very low and near zero, which means the team did not decrease or increase the cyber mission capability rate over the course of the simulated cyber conflict. This is expected due to the similar level of capabilities presented by both the attacking and defending cyber teams.

4.9.11 Cyber Operational Efficiency Results

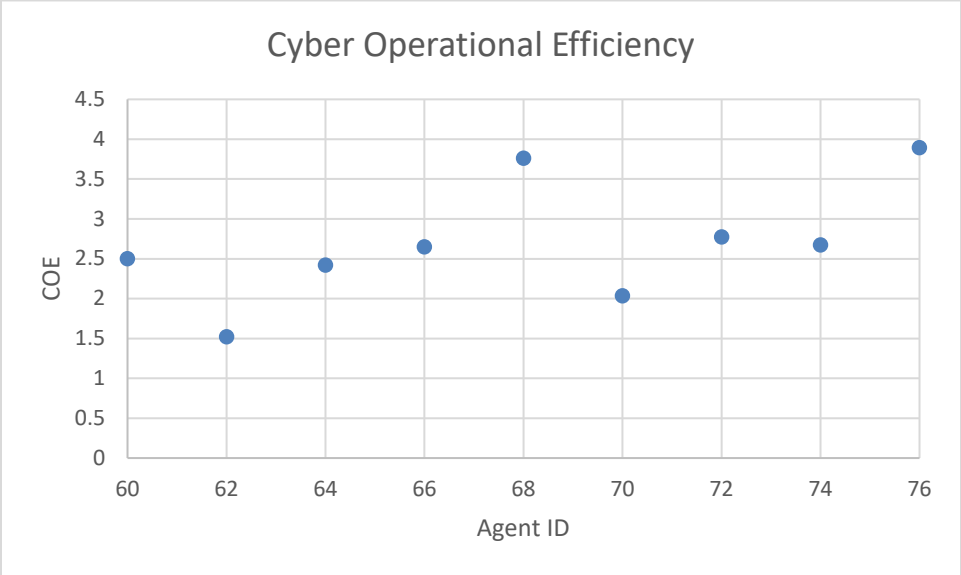


Figure 33: Agent Cyber Operational Efficiency

4.9.12 Cyber Operational Efficiency Discussion

The cyber operational efficiency performance measures simulated for each agent fell in line with what would be expected based on their knowledge, skill, and experience. In this simulation, the team was made up of eight hosts or network squad members, with knowledge, skill, experience (KSE) values of either all one, all two, or all three. These are agents 62 – 76. Agent 60 is the team lead, so while efficiency is tracked, it is not meaningful when comparing and contrasting with the other agents because the team lead

tasks are abstracted into operations related to communication and management. Agents 62 – 76 are conducting survey and secure operations where they are actively searching for vulnerabilities, attempting to remove vulnerabilities, and attempting to restore compromised terrain when alerted. The resultant cyber operational efficiency measures, as shown in Figure 33, for each agent are lowest for KSE 1 (agents 62 and 70), middle for KSE 2 (agents 64, 66, 72, and 74), and highest for KSE 3 (agents 68 and 76). Taken altogether, as a team, the average cyber operational efficiency is 2.694. This value, by itself, is meaningless. Team cyber operational efficiency becomes meaningful when simulations containing different teams of varying size and KSE values are run, and then compared against one another.

4.9.13 Compromise Success Rate and Time to Compromise Results

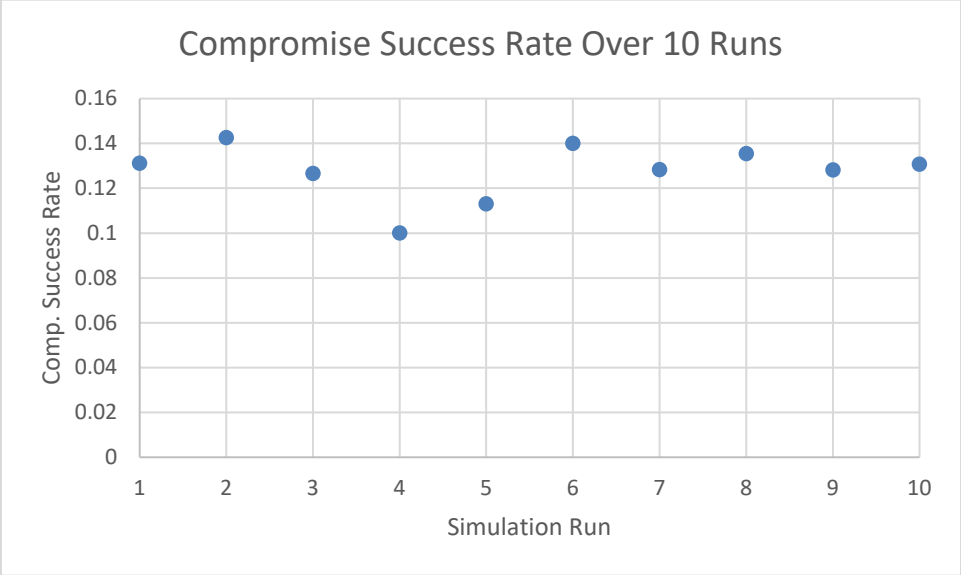


Figure 34: Attacker Agent Compromise Success Rate

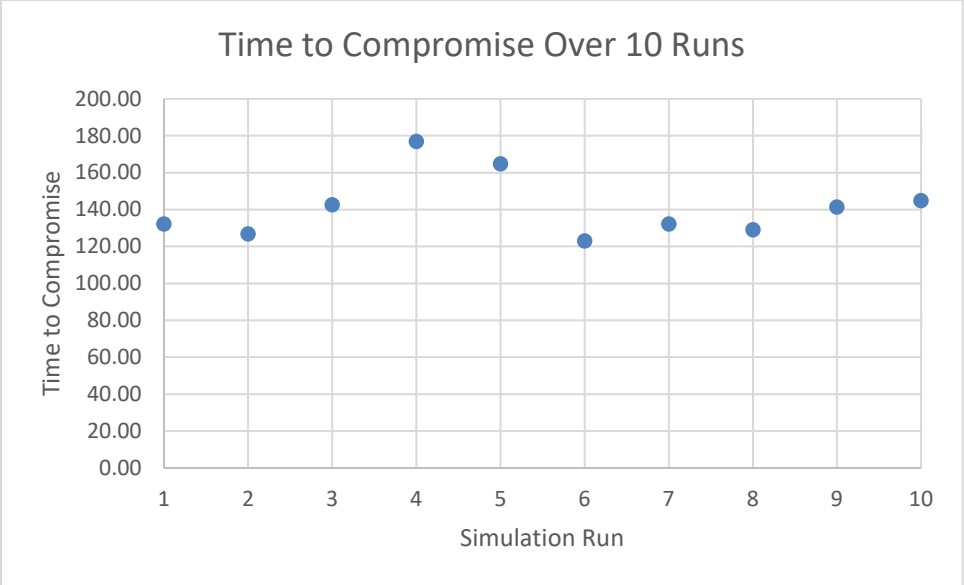


Figure 35: Attacker Time to Compromise

4.9.14 Compromise Success Rate and Time to Compromise Discussion

Compromise success rate and time to compromise are both measures of how well the attacking team is doing, and therefore the defending team (the focus of this model) is aiming to minimize the former and maximize the latter. Figure 34 shows the comprise success rate over the simulation runs. The minimum compromise success rate simulated was 0.100, the maximum was 0.143, and the average for all runs was 0.128. Figure 35 shows the time to compromise over the simulation runs. The minimum time to compromise was 123.099, the maximum was 177.024, and the average for all runs was 141.42. This data would be extremely difficult to compare with real world operations due to the limited information available at successful attacks. For compromise success rate, it would seem that the simulated values (approximately 0.128) are somewhat reasonable but likely higher than real world. Also, it would have to depend on the real-world definition of success. In the Cyber-FIT model, the denominator includes all attempted attacks where the attacker agent attempts to deliver payload. In real world that could be expanded to starting with reconnaissance operations or limited to only once payload is delivered.

4.9.15 Network Measures Results

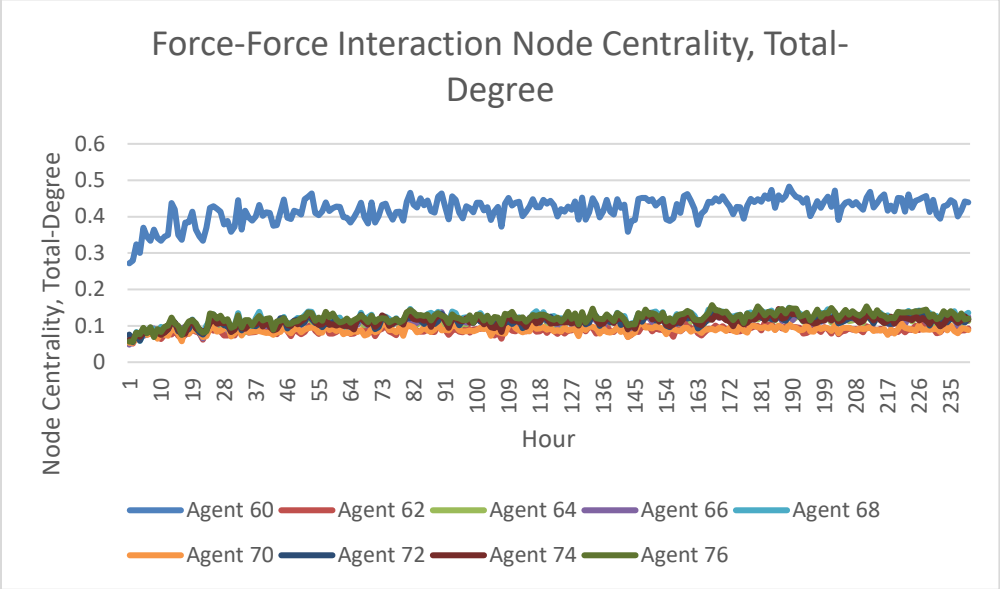


Figure 36: Force-Force Interaction Node Centrality, Total-Degree

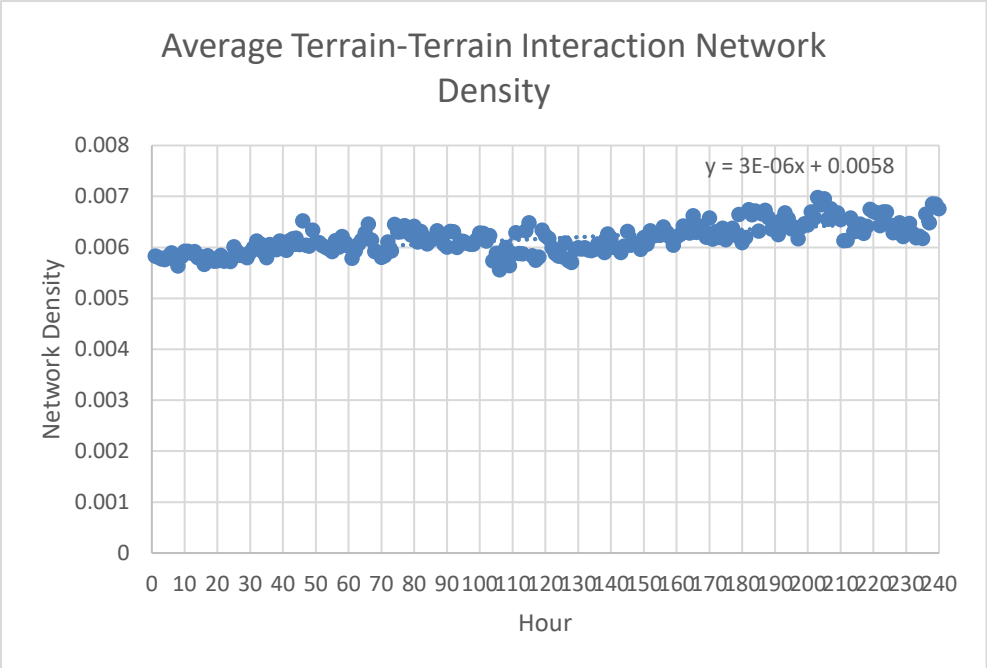


Figure 37: Terrain-Terrain Interaction Network Density

4.9.16 Network Measures Discussion

Two network measures were selected to collect and then use for dynamic network analysis. Using network measures for tracking and interpreting cyber team performance is

less direct than the previous measures discussed. That is, there is no specific value a team would be aiming for in terms of network measures to gauge cyber team performance because not enough is known yet. Whereas, in the case of terrain vulnerability rate, the team is clearly working towards the lowest value possible, ideally zero. This means that in the case of network measures, trend analysis and over time correlation would be more appropriate. The dynamic network analysis for both measures was computed using ORA. Figure 36 shows a node level measure (centralization, total-degree) calculated on the collection of links at every hour. Clearly, agent 60, the team lead, has the highest node centralization total-degree during the entirety of the simulated conflict, which is expected. The other agents, on average, have similar values that vary within a small range throughout the simulation. Since the current version of Cyber-FIT doesn't have a wide range of behaviors, the agents will behave similarly. These two network measures are provided in this version as a proof of concept, which is shown to work at a basic level. Figure 37 shows the terrain-terrain interaction network density dynamic network analysis. The best fit linear curve to the average terrain-terrain interaction network density is shown with a slope of 0.000003. There is a very small increase over time, likely due to the slight increase in the number of vulnerabilities and compromises throughout the simulation, causing more activity amongst the team (which increases interactions amongst terrain being used for surveying and securing operations). Frequently, network visualization is coupled with network analysis to get a better sense of what is occurring. Figures 38 and 39 show a network picture of a randomly selected hour of the dynamic network analysis, produced by ORA.

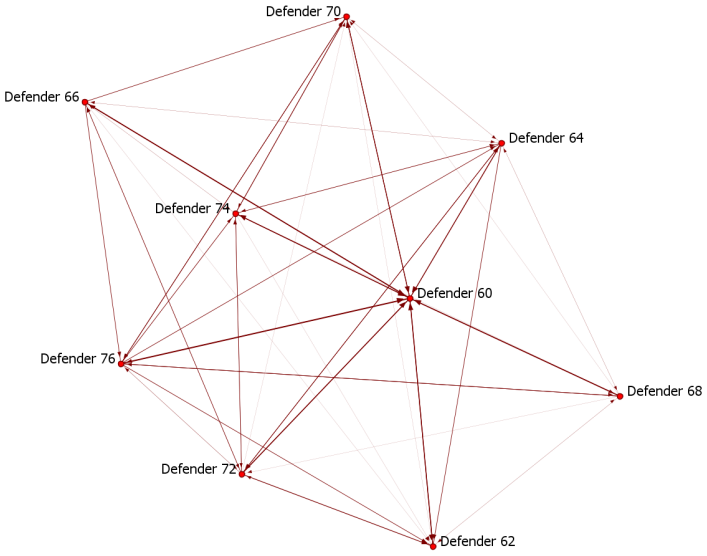


Figure 38: One hour visualization of Agent-Agent Link Network Node Centrality, Total-Degree

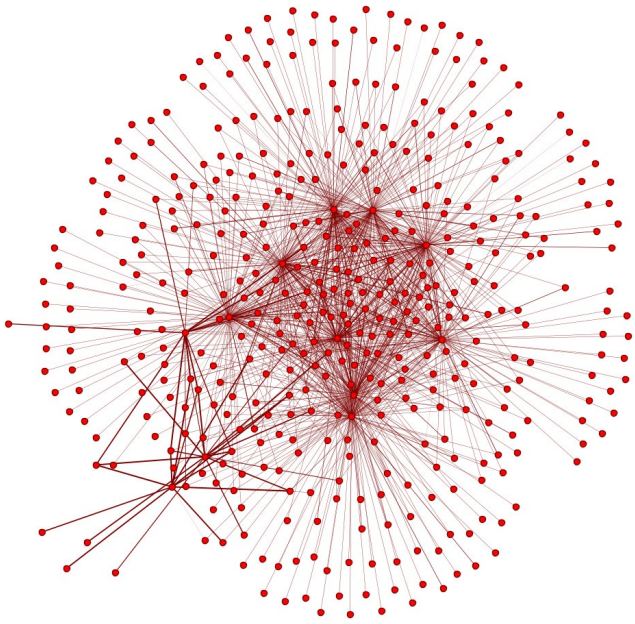


Figure 39: One hour network visualization of Terrain-Terrain Network Density

4.9.17 Mission Defined Measures Results

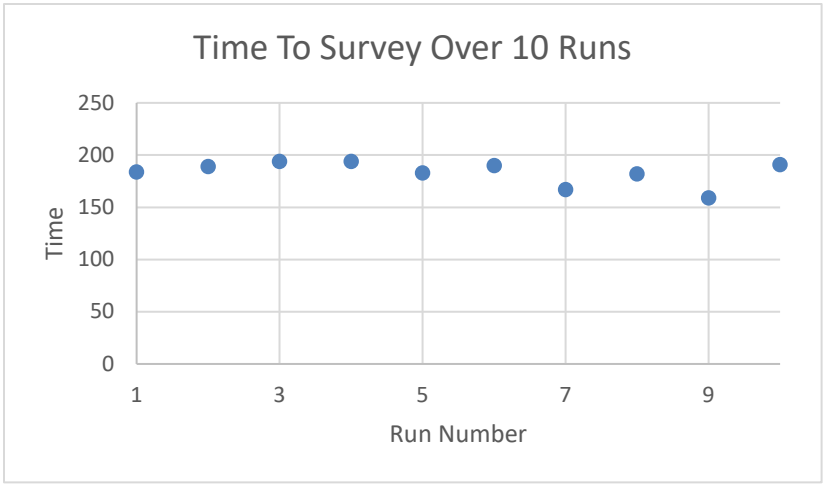


Figure 40: Time to Survey

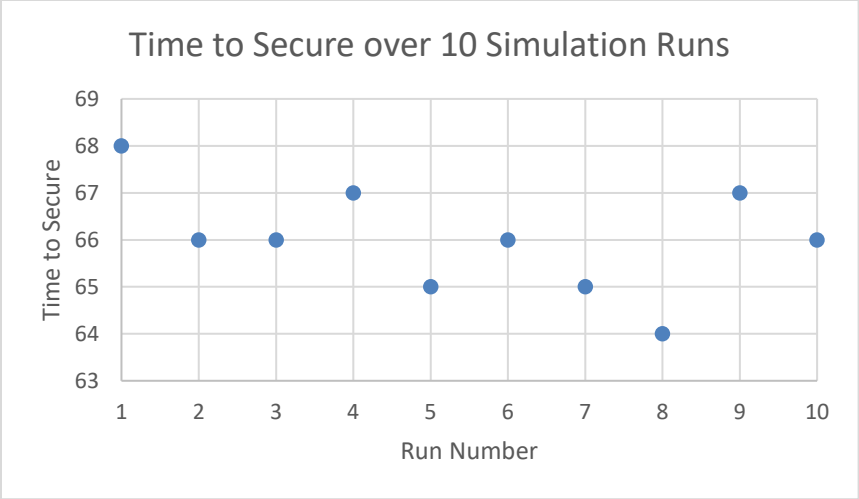


Figure 41: Time to Secure

4.9.18 Mission Defined Measures Discussion

As shown in Figure 40, the ten simulated missions resulted in the cyber team completing its survey mission between 159 minutes and 194 minutes. Figure 41 shows the ten simulated missions resulted in the team completing its secure mission between 64 and 68 minutes. Both of these missions were simulated with a two-day pre-deployment time, and then an eight-day mission with no attacker agents present (all other simulation setup variables were the same). This resulted in both measures having very little variance as can be seen in both figures. Since these two measures are mission-defined they can be set to include more complexity, which would result in more interesting results. For instance, the cyber team could secure terrain for a specific amount of time, achieving a very low terrain vulnerability rate, before attacker agents begin their operations, to see how low the team can keep the terrain vulnerability rate, with an active opposing force. Similarly, time to survey could be altered to include only the highest tier vulnerabilities (most severe and concerning) or only systems supporting the most important missions. This is exactly why leadership defines mission parameters in real world operations, because it is situational to what is occurring at what priority.

4.9.19 Team Performance Dashboard

A key motivation to this software simulation framework is helping move the state of the art in the direction of a comprehensive view of cyber team performance. Cyber-FIT Version 4 generates data in the form of comma separated value files reporting agent level data. These data are then post-processed and plotted into charts that were displayed in the previous section. Collectively, these charts can serve as a prototype of cyber team performance dashboard. The previously discussed Defense Science Board report [28],

displayed a notional consideration of what a system performance dashboard should look like. At the time, none of those measures were formally defined. Cyber-FIT Version 4 defines the measures of performance, embedded in software, and then programmatically simulates and computes them. The Defense Science Board notional dashboard shown in Figure 42 can be compared and contrasted with the dashboard provided by Cyber-FIT shown in Figure 43.

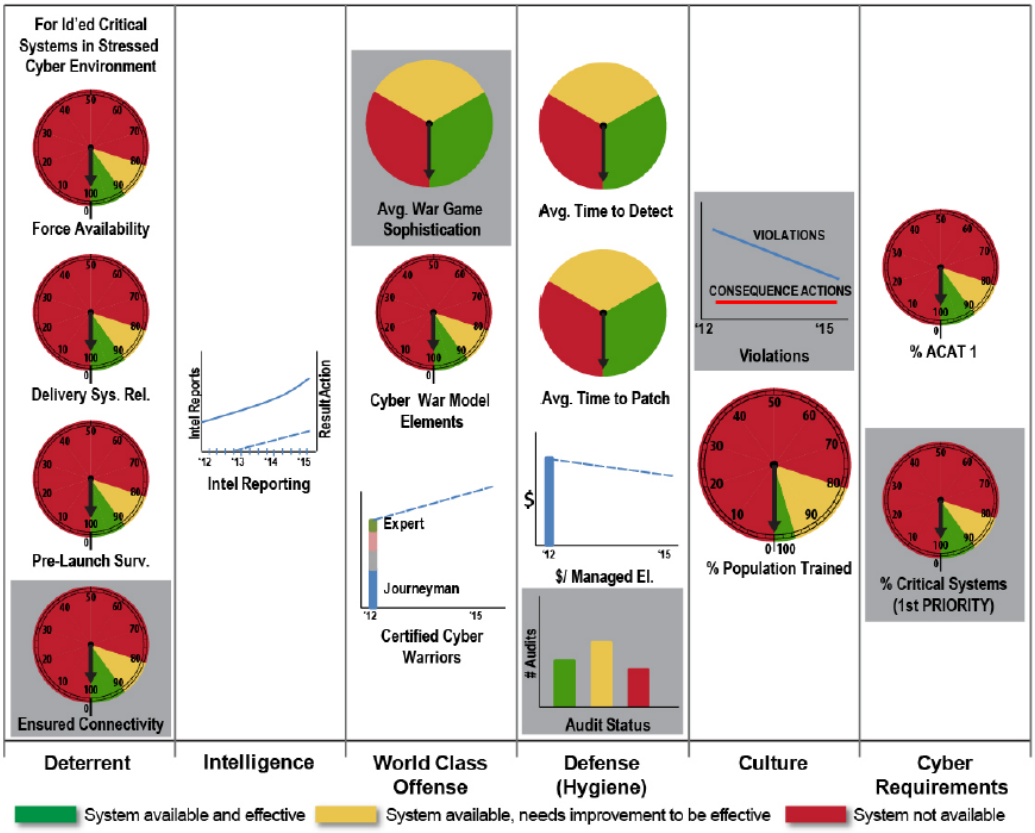


Figure 42: Defense Science Board Notional Cyber System Performance Dashboard

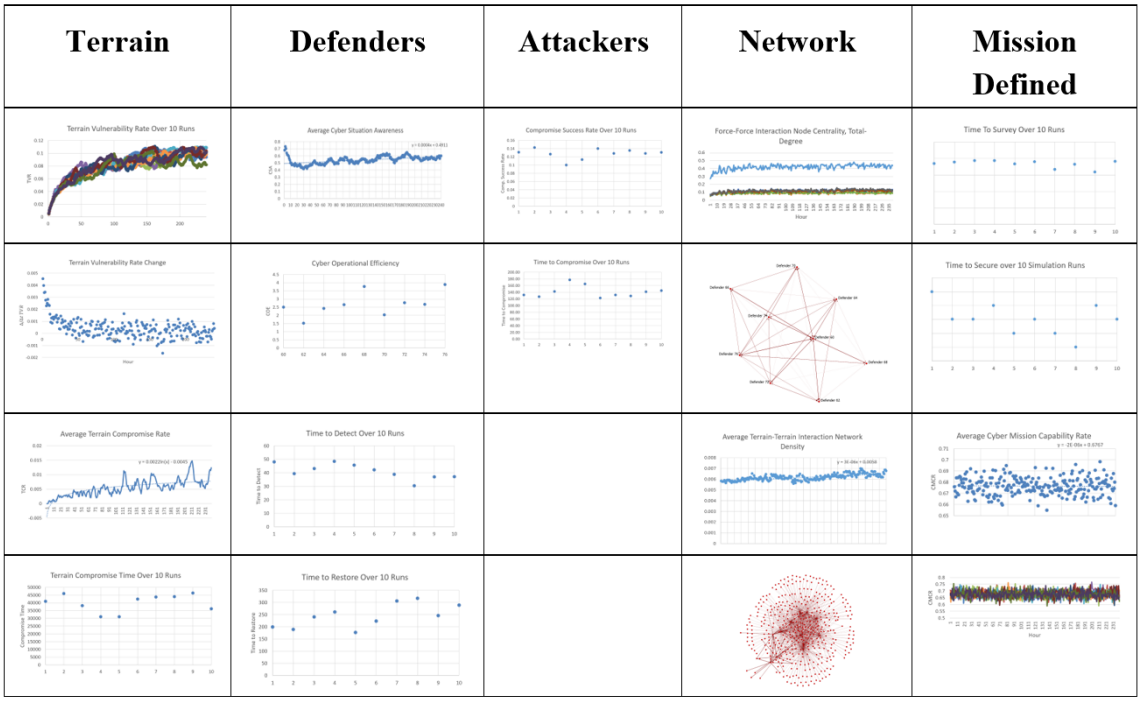


Figure 43: Cyber-FIT simulated cyber team performance dashboard

First, consider the categories of each. The notional dashboard breaks up the measures by: deterrent, intelligence, world class offense, defense, culture, and cyber requirements. Clearly, this dashboard is of a higher scope than the Cyber-FIT dashboard, which computes and displays team performance measures only. The design of the Cyber-FIT dashboard was based on the agent type aggregation of measures, which leads to fairly straight forward categories to group the measures: terrain, defenders, attackers, network and mission defined. Terrain measures are reporting on the cyber systems specifically – how vulnerable, available, and compromised they are. The defender category reflects the operational performance of the cyber forces tasked with defending the terrain, while the attacker category is the reverse of that. The network category provides network centric measures of interactions occurring on both the cyber systems and forces. These measures would have to be calibrated over time so change detection techniques can be utilized. Finally, the mission defined category would be set by leadership to track the measures specifically set by commanders. In comparing both dashboards, there are eight measures having a strong similarity between what was notionally proposed and what is being computationally modeled and simulated in Cyber-FIT: force availability, pre-launch survey, certified cyber warriors, average time to detect, average time to patch, audit status, percent ACAT 1, and percent critical systems. These are not perfect representations between dashboards, but a close enough approximation to fulfil some level of the vision proposed by the Defense Science Board.

4.10 Model sensitivity analysis

Cyber-FIT version 4 is a stable model with respect to the main development goal of outputting cyber team performance measures in a realistically scaled simulation. Arriving at this point took a considerable effort in terms of iterative development and model calibration. As defined by Carley, calibration is the "process of tuning a model to fit detailed real data. This is a multi-step, often iterative, process in which the model's processes are altered so that the model's predictions come to fit, with reasonable tolerance, a set of detailed real data" [77]. The detailed set of data in this case are the cyber team performance measures. There are several key control variables that were calibrated in order to arrive at the response surfaces detailed in this chapter. These control variables should exert influence over the outcome variables, and to test this a one-at-a-time sensitivity analysis in the form of parameter sweeping is conducted. Each of the control variables used for sensitivity analysis are the most influential agent class variables: restoral rate (defender agent), vulnerability growth rate (terrain agent), and exploit rate (attacker agent). For all sensitivity analysis simulation runs, all other control variables were held constant. This ensures that no other control variable is affecting the parameter under consideration for a particular sensitivity analysis.

The current version of Cyber-FIT is not expected to match reality. Much of this thesis, and related work, is showing how difficult it is to define, much less simulate cyberspace reality. This work is early research in an emerging phenomenon and aims to create a framework from which more realistic analysis can emerge. This principle guides the sensitivity analysis. Parameter sweeps of the most important control variables are aimed at showing that outcome measures are highly sensitive to changes. This will show that these are the key control variables holding the model together and keeping it from completely erratic, or meaningless behavior.

For all runs of the sensitivity analysis across all variables the same mission, cyber defender and cyber attacker configurations were loaded. This configuration is a realistically relevant conflict setup with hundreds of cyber terrain, one cyber defending team and two cyber adversaries. The following table details these configuration settings for sensitivity analysis in the following sections.

Cyber Terrain Configuration				
Mission	Friendlies	Networking Terrain	Server Terrain	Host Terrain
Base	0	10	20	30
1	10	2	3	15
2	20	3	4	25

3	50	5	5	60
4	50	5	5	60
5	100	6	12	130
Defender Agent Configuration				
ID	Squad	Knowledge	Skill	Experience
1	1	2	2	2
2	2	1	1	1
3	2	2	2	2
4	2	3	3	3
5	3	1	1	1
6	3	2	2	2
7	3	3	3	3
Attacker Agent Configuration				
ID	Tier			
1	3			
2	4			

Table 25: Sensitivity analysis mission, defender, attacker configuration summary

4.10.1 Terrain agent vulnerability growth rate sensitivity

Vulnerability growth rate is the rate at which vulnerabilities manifest on cyber terrain agents. In real world operations vulnerability management is one of the primary functions of any corporate information technology (IT) department. A vulnerability is a "conditions or behaviors that allow the violation of an explicit or implicit security policy" [78]. Vulnerabilities can be of many forms like software, hardware, firmware, logical, and networking. IT departments use vulnerability management software to monitor systems for the presence of known vulnerabilities. There is also an ecosystem that finds, reports, and patches vulnerabilities, with different incentive structures in place for various players. Software companies want to find vulnerabilities so that their software can be patched and considered safe, so they will often pay bounties for vulnerability discovery [79]. On the flip side, criminal organizations also pay bounties for vulnerability discovery so that the companies aren't able to patch. These vulnerabilities are exploited by malicious actors. These vulnerabilities are often referred to as "zero-day" exploits because it has been zero

days since discovery and/or remediation. The figure below shows the CERT process for Coordinated Vulnerability Disclosure.

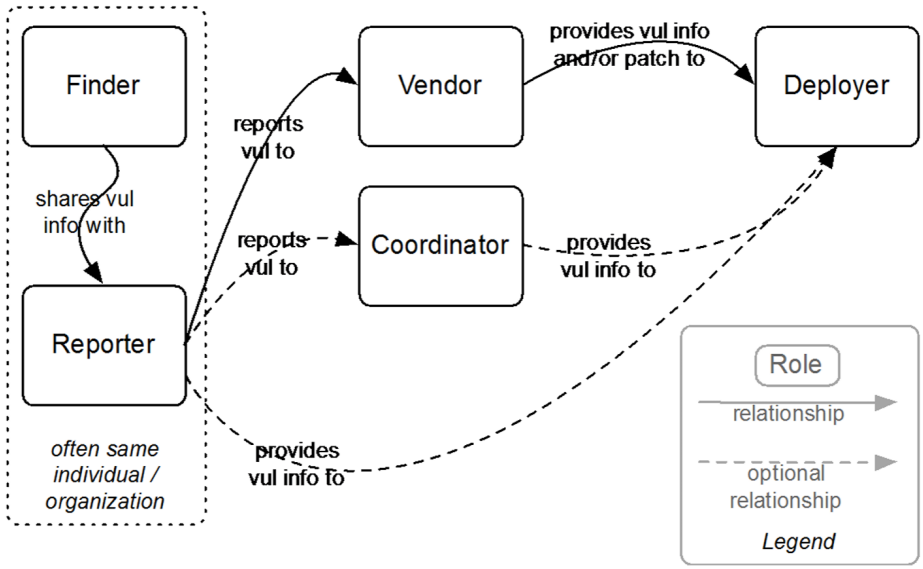


Figure 44: The CERT process for coordinated vulnerability disclosure

Cyber-FIT abstracts this real-world phenomenon into a rate at which vulnerabilities appear on cyber terrain. At any given time, each terrain agent has a list of vulnerabilities present. The vulnerabilities can be any integer from [0 – 99]. The value represents the complexity, with the highest number the most complex and difficult vulnerability to discover, patch, and exploit. The only exception is vulnerability number 0, which represents a zero-day and can only be exploited by Tier 6 attacker agents. A zero-day is a representation of the time between a vulnerability emerges, and the time the Deployer (in the Figure above) is able to publish the vulnerability details. Vulnerability growth rate (VGR) parameter sweep details and results are shown in the tables and figures below.

Parameter:	Vulnerability growth rate (VGR)
Values:	[1, 0.1, 0.01, … 0.0000001]
Outcome variable:	Total Vulnerability Rate (TVR)
Runs per setting:	10
Total runs:	70

Table 26: Vulnerability growth rate sensitivity analysis settings

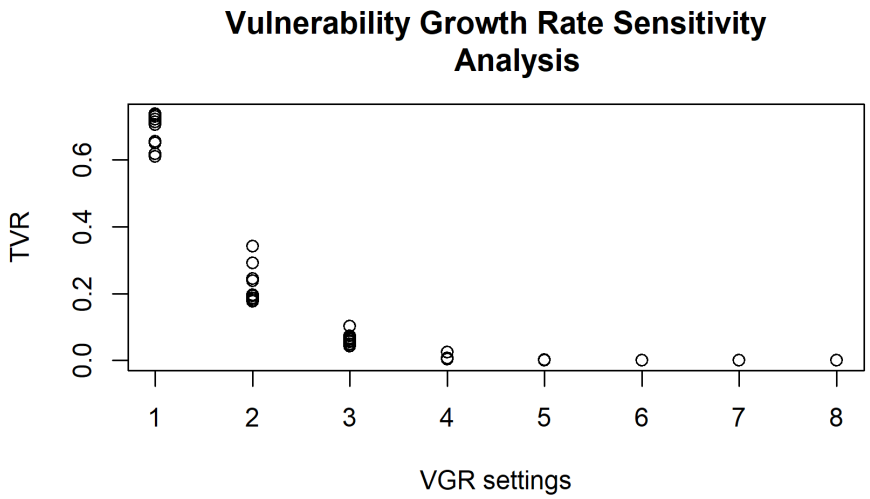


Figure 45: Terrain vulnerability rate sensitivity to vulnerability growth rate

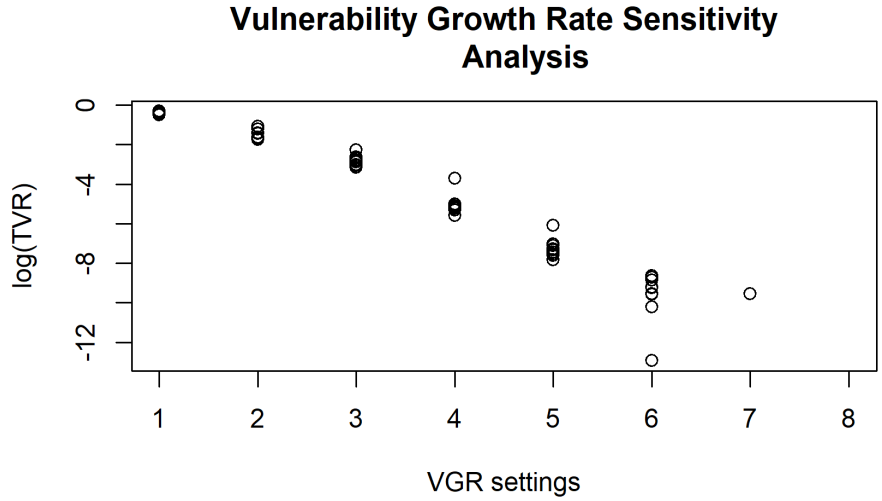


Figure 46: Logarithmic curve of terrain vulnerability rate sensitivity to vulnerability growth rate

Setting	Mean TVR	Standard Dev.	Range
1	0.68727	0.04675	[0.609 – 0.738]
2	0.22252	0.05338	[0.176 – 0.341]
3	0.06157	0.01632	[0.042 – 0.103]
4	0.00750	0.00578	[0.0037 – 0.0247]
5	0.00081	0.00050	[0.0004 – 0.0023]
6	0.00008	0.00006	[0 – 0.0002]
7	0.0	0	[0 – 0]

Table 27: Terrain vulnerability rate sensitivity to vulnerability growth rate summary statistics

Altering VGR has a large effect on the total vulnerability rate (TVR). This is expected behavior and seems to track to real world expectations. A terrain vulnerability growth rate of 1 means that a new vulnerability emerges every minute of every day on every computer in the network, which is clearly not realistic. However, if the vulnerability growth rate is too low, close to zero vulnerabilities will ever emerge, and those that do are easily removed by an active cyber team. This response surface is shown in the two figures above. The first shows terrain vulnerability rate decreasing for each setting (where each setting is decreasing the VGR by a factor of 10). Since the decrease is so rapid, a log plot of the terrain vulnerability rate in the second figure shows a linear gradient until setting seven where only one sample is above zero and then setting eight (VGR = .0000001) where all simulation runs result in an average TVR of zero.

So what TVR is ideal? There is no specific answer to this question and this is more art than science. Obviously setting one is too high and setting seven is too low. There is no existing literature on how often systems become vulnerable or how many vulnerabilities per system. Existing research does tell us that there are approximately 6,000 software vulnerability discoveries per year [80]. So the question becomes how many vulnerabilities are present on any one machine at any given time. The realistic answer is probably a few. This is because well maintained computer networks are constantly patched and updated (what the cyber team in Cyber-FIT is simulating). Therefore, zero up to several vulnerabilities at any given time represents a vulnerability level of approximately 0.01. In Cyber-FIT the total vulnerability space is all integers 1 – 99 (summing to 4,950). So if for example there is one vulnerability, level ten then that system vulnerability level is 10/4,950 or .002. If there are several vulnerabilities totaling 120, then the system vulnerability level is .024. Therefore, setting four is the setting which most closely resembles a reasonable expectation (VGR = 0.001). More importantly, once this setting is selected, and coupled

with others going through this selection process of tuning, the overall simulation dashboard resembles values we'd reasonably expect in a security operations center.

4.10.2 Defender agent restoral rate sensitivity

When it comes to dealing with compromised systems, the speed at which a cyber team member can remediate a problem is likely the most important skill to have. Security operations centers primary function is to ensure computing systems are operating normally. Normal can be defined as responding to informational inputs, making computations, and then storing and outputting data. In reality, this becomes very complex and will manifest in different ways for different systems. A switch provides paths for internet protocol software lookups. A server hosts web pages that respond to requests. In cyberspace operations, definitions of normal could span an enormous range of options.

Cyber-FIT version 4 tracks each terrain agent as operating normally or compromised. An additional complexity is added where the terrain agent vulnerability level affects how quickly the terrain agent will output information when requested. For the purpose of this sensitivity analysis, only compromised versus operating normally is considered. Clearly if the restoral rate is zero, no system will ever be restored and if the restoral rate is 1.0 then every system will be immediately restored upon a defender agent interacting with it. For each of these extremes, either an extremely high total compromise time will be output or an extremely low total terrain compromise time.

Empirical data on system compromise time is not readily available. The reason for this is likely two-fold. First, firms are reluctant to disclose system compromise data because it might tarnish their reputation and/or embolden attackers. Second, for large networks its often difficult to quantify effects of cyber attacks and also differentiate benign system malfunctions. A legitimate attack may be discovered, but there may be uncertainty with that particular attack's reach into the network. There are empirical studies on system compromises in the form of categorizing and defining attack characteristics [81] and analyzing timing issues of when attacks are detected [82].

Using compromise time, even though not well known empirically, is a simple and effective measure for what the defender restoral rate should most affect in cyber team performance model. In this version of Cyber-FIT, defender agents have knowledge, skill, and experience characteristics. How those traits combine or interact to determine how well an agent will do when fixing compromised systems is of current research interest and the subject of a virtual experiment later in this chapter. For this sensitivity analysis, skill alone will determine what the restoral rate is for a defender agent. Since skill can be low, medium, or high, the settings for this sensitivity analysis will take that into account. The following table lists the restoral rates for each skill level over all settings of the sensitivity analysis.

Setting	Low Skill	Medium Skill	High Skill
1	1	1	1
2	.1	.5	1
3	.01	.05	.1
4	.001	.005	.01
5	.0001	.0005	.001
6	.00001	.00005	.0001
7	.000001	.000005	.00001
8	.0000001	.0000005	.000001

Table 28: Defender agent restoral rate settings for sensitivity analysis

These eight settings were run ten times each. The following table describes the sensitivity analysis details.

Parameter:	Defender restoral rate
Values:	Table 28
Outcome variable:	Compromise time
Runs per setting:	10
Total runs:	80

Table 29: Sensitivity analysis set up table

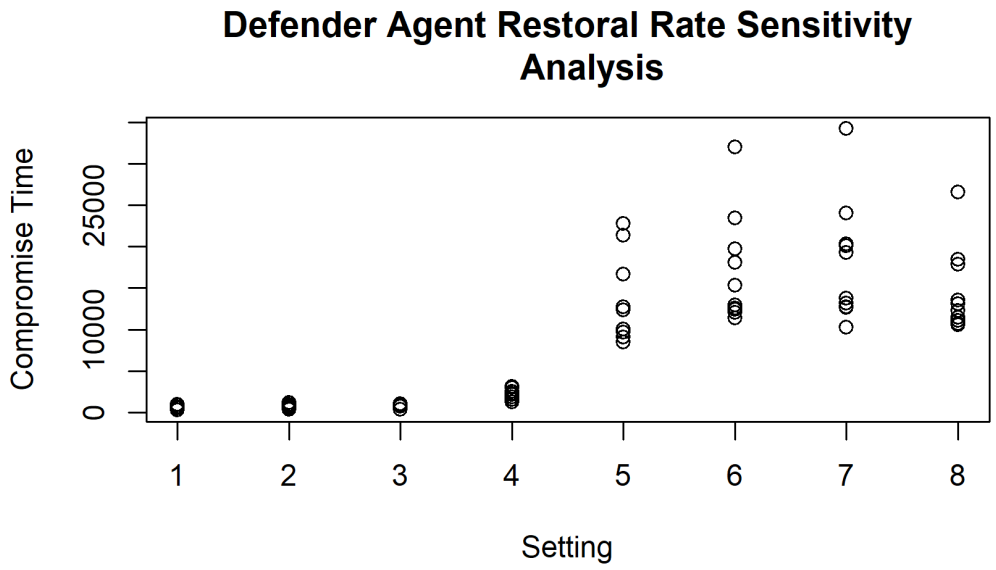


Figure 47: Defender agent restoral rate sensitivity results

As shown in Figure 47, the outcome measure of compromise time is highly sensitive to the defender restoral rate. With very high restoral rates, compromised systems are quickly remediated. This holds for settings one through three. At setting four the compromise time moves up and more variance is shown. At setting five is where the compromise time jumps greatly. It seems that anything lower than setting five does not make much difference, this means that the restoral rate is sufficiently low that the compromised terrains will rarely be restored. More detailed results of the sensitivity analysis are shown in the table below including range and standard deviation. The sensitivity analysis shows that restoral rate has a high impact on the compromise time, meaning the model is working as expected. More importantly, this type of simulation provides hypothesis generation. For example, based on this model, what combination of settings between skill levels is most realistic? What effect happens when more forces are added or more terrain is in play? Many more detailed modeling and analysis can be developed using Cyber-FIT.

Setting	Mean Comp. Time	Standard Dev.	Range
1	576.8	201.7	[238 - 981]
2	627.1	248.2	[356 – 1,171]
3	792.9	170.2	[341 – 1,002]
4	2,102.4	592.9	[1,227 – 3,108]

5	13,330.8	4,934.4	[8,460 – 22,783]
6	17,004.2	6,269.5	[11,391 – 32,054]
7	18,057.6	6,854.4	[10,279 – 34,269]
8	14,576.2	4,823.2	[10,584 – 26,618]

Table 30: Restoral rate sensitivity analysis all settings summary statistics

4.10.3 Attacker agent exploitation success rate sensitivity

Modeling the behavior of cyber attackers has been an active research subject over the recent years. Several attempts at defining these behaviors into taxonomies have been made [83] [84]. There still is not a universal definition or standard that can define cyber attacks to ingest into a software tool. Work of this nature might eventually lead to a standard published by the IEEE or NIST. Separately from taxonomies there is much research about the details of different types of cyber attacks. For instance, since computers are connected through networks, graph analysis of how attacks manifest has proven to be effective in managing and visualizing cyber attacks [85]. Another methodology focused on using attack vectors to predict computer network vulnerability level [86]. This type of approach is one of the "red teaming" techniques. Red teaming is when a group that is part of the organization mimics adversarial behavior to find problems that might be exploited. There are examples of empirical studies of computer intrusions and failures in the literature. For example, work analyzing a large failure data set from two different high-performance computing sites [87]. This study primarily sought to find patterns in the data. Another study [88] used various datasets like the previous study to analyze distributions of anomaly data. Both papers mention the difficulty to pinpoint realistic descriptions of failures and the many possible confounding data. These are not clean, labeled datasets to work with.

Overall, predicting how cyber attackers will ply their craft continues to be very difficult. The frequency of cyber attacks continues to rise all over the world. According to a 2019 Accenture report [89], cyber attacks have increased 11% since 2018 and 67% since 2014. Militaries are confronting more sophisticated adversaries and continue to evolve their cyberspace operational forces. Cyber attacks are usually designed to happen fast, causing damage to machines, infecting other machines, and taking objective actions, before covering tracks and moving on. This behavior has been well established by various frameworks probably most notable the Lockheed Martin Cyber Kill Chain, depicted in the next figure.

Figure 48: Lockheed Martin Cyber Kill Chain [47]

The cyber kill chain behavior has been modeled into Cyber-FIT since version two for several purposes. First, adds temporal realism. The attacker agents cannot simply start infecting machines and shutting down servers. They must spend the time needed to get through the cyber kill chain upon beginning an attack. With that in mind, virtual experiments were conducted showing which defensive maneuvers might affect timing issues moving through the cyber kill chain. Secondly, the specific behaviors that leave signals within the cyber terrain can be defined. Unusual scanning activity might tip off the cyber team that reconnaissance activity is occurring. Unusual machine-to-machine communications could be an indicator of command and control. In any event, this type of attacking agent behavior has to be designed, developed, simulated, and tested in order to increase our awareness in how adversarial behavior looks in the real world.

As described previously, the attacker agents in Cyber-FIT version four work through the cyber kill chain. During reconnaissance, they search for vulnerabilities present on the cyber terrain agents. During this time, they build a list of found vulnerabilities. Next, they weaponize on vulnerabilities that their tier level allows them to. Once a list of attacks has been built, they attempt to deliver these attacks to terrain agents. At this point the exploitation and installation phases occur, stochastically simulating success. Next, if successful, time passes while the command and control and actions on objectives phases play out. The critical control variable for the attacking agent's overall behavior, is the exploit success rate. That is, during the exploitation phase, does the attack, matching a

vulnerability, work. This control variable is the subject of attacker agent sensitivity analysis. The sensitivity analysis conditions are shown in the following table.

Parameter:	Attacker exploitation success rate
Values:	[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9.1.0]
Outcome variable:	Attack Success Rate
Runs per setting:	10
Total runs:	90

Table 31: Sensitivity analysis setup

The sensitivity analysis is run for all conditions. Attack success rate is shown to be sensitive to exploit success rate. The following figure shows a box plot of all runs of the sensitivity analysis.

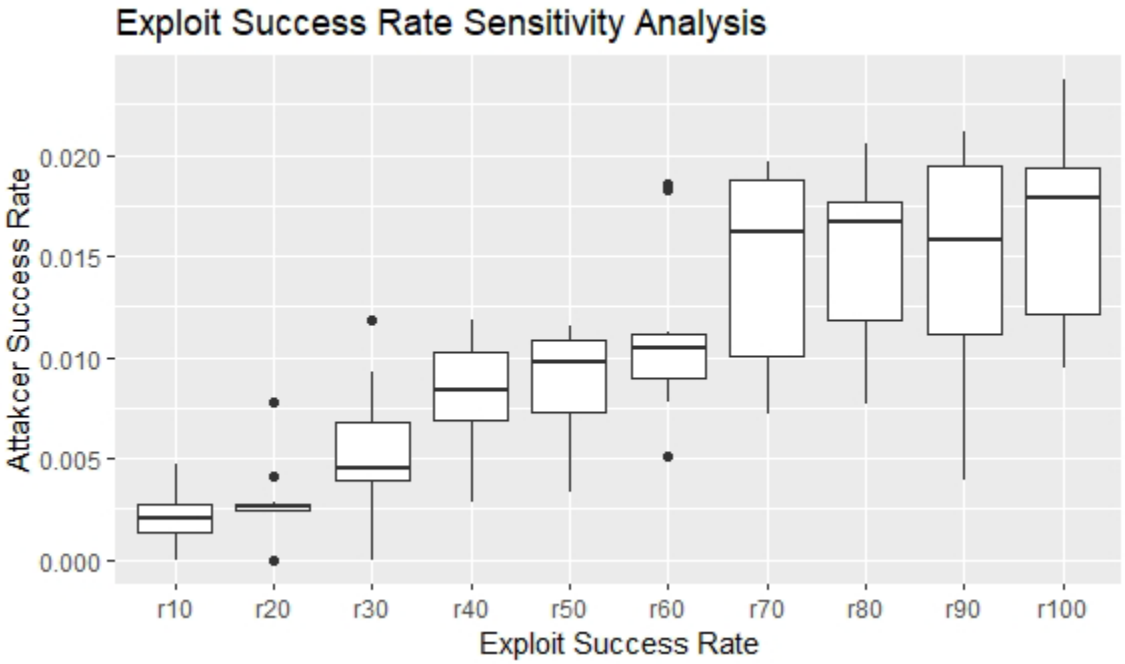


Figure 49: Exploit success rate sensitivity analysis

As shown in Figure 49, as exploit success rate increases, the attacker agent exploit rate increases until 0.7. For values of 0.7, 0.8, 0.9, and 1.0, the attacker exploit rate is the same. This shows that the model has a hard boundary at this point. For an attacker's exploit to be successful, several things must happen. First, the attacker must find a vulnerability to exploit. Then time will pass as the attacker agent weaponizes, and then delivers the exploit to a target terrain agent. At this point, during the exploitation phase,

the exploit has to be a matched to the vulnerability it targets. Two possibilities for failure may occur at this point. The vulnerability may have been removed due to defensive cyber operations by the defender agents, or the vulnerability never existed on this terrain agent (because the attacker agent delivered this exploit to the wrong target). In either of those two cases, the attacker agent learns that the attack attempt failed. Those two cases make up a large portion of the potential outcomes. The success condition occurs when the exploit is delivered to a terrain agent that currently has the targeted vulnerability and the random variable draw is less than the exploit success rate control variable. This process is depicted as a flow chart in the following figure.

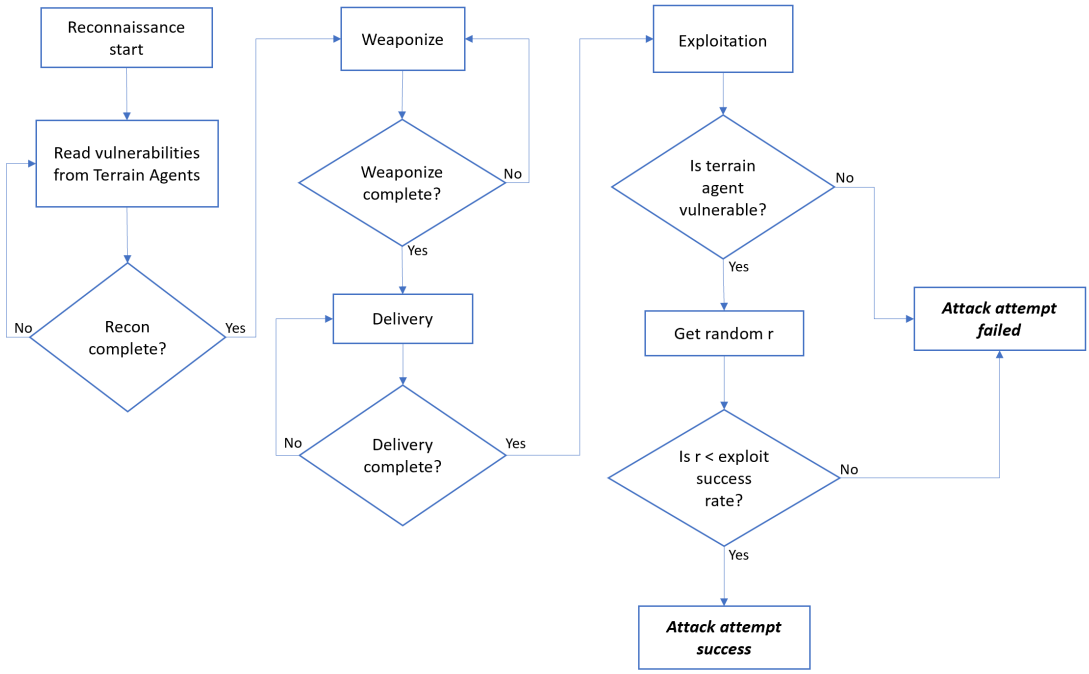


Figure 50: Attacker agent flow chart to attack success based on exploit success rate

As mentioned, there are two ways that the attacker could be in the exploitation phase with a specific terrain, and that terrain not being vulnerable to that exploit. If that is the case the attack has failed. Or, the random number draw, r, could be greater than the exploit success rate, which would also result in the attack attempt failing. There are many times when the former occurs, which is difficult to predict and can only be realized by building the model and then running sensitivity analyses. This analysis shows that, under the current settings, the attacker exploit rate has a hard boundary of approximately 0.025. All runs of the sensitivity analysis are summarized in the table below, showing the range and mean of each setting.

Exploit Success Rate	Range	Mean
0.1	[0.0000 – 0.0047]	0.0021
0.2	[0.0000 – 0.0078]	0.0028
0.3	[0.0000 – 0.0118]	0.0053
0.4	[0.0028 – 0.0118]	0.0082
0.5	[0.0033 – 0.0115]	0.0088
0.6	[0.0051 – 0.0185]	0.0111
0.7	[0.0072 – 0.0196]	0.0144
0.8	[0.0077 – 0.0205]	0.0151
0.9	[0.0039 – 0.0211]	0.0148
1.0	[0.0095 – 0.0237]	0.0165

Table 32: Exploit success rate all runs range and mean

One important consideration is the definition of attacker success rate drives these values. In this model, all attempts by an attacker include all delivery attempts. In reality, an attacker might attempt to delivery payload to a number of computers, and if just one of those computers become exploited, the purpose of the attack could be considered successful. If Cyber-FIT calculated success in that way, the exploit success rates would be much higher. Also, it's hard to know what realistic attack success rates are. Empirical data is extremely hard to come by on this subject. There is existing work in creating detailed attack paths and building probability models of those paths being successful [90]. Work like this will be informed by some empirical data in the form of surveys and publicly available internet data. As of this writing, there hasn't been research using empirical data on how successful specific cyber attacks are likely to be. So, in this current version, with the simulation set up as described, the attacker success rate ranges from 0.000 to 0.024. Is this realistic? Nobody knows, and work will have to progress in this field to move to a closer understanding of what's actually occurring in the real world. For the purpose of a working model that results in realistic cyber team performance measures, these values work fine.

4.11 Virtual experiments

One of the primary use cases for an agent-based software like Cyber-FIT is virtual experimentation. In fact, each version of Cyber-FIT development followed the same pattern: identify research question(s), model the most important behavior affecting the research question, and then run virtual experiments. This pattern is useful because it

focuses the added complexity to a timely and interesting aspect of cyber warfare modeling and simulation. Cyber-FIT version 4 is focused on the additional complexity of defender agents. This additional complexity takes the form of segmenting squad, which systems to interact with, and how knowledge, skill, and experience differentiate outcomes. This differentiation is of active interest to the military because of decisions that must be made in terms of policy and resource management. Recently, General Paul Nakasone, the commander of United States Cyber Command, and director of the National Security Agency, made remarks to Congress about on this subject, saying: "Training and proficiency are improving through our mission simulation capabilities, particularly the Persistent Cyber Training Environment (PCTE). The PCTE is helping us mature cyber operations tradecraft, enhance individual proficiencies and enable faster attainment of team certification and collective training in maneuvers such as Exercise CYBER FLAG" [91]. Exercise CYBER FLAG is one of the biggest cyber war exercises in the world, consisting of dozens of teams across more than 20 countries all working to improve their cyber operational skillset [92]. This training is enormously expensive in large part because it is extremely difficult. The fiscal year 2023 cyber force size is slated to increase by adding cyber teams costing approximately $2.5 billion to fill, train, and equip [93]. So, the question is: once the teams are filled – how should they be trained?

4.11.1 Virtual experiment one design

When envisioning a perfect cyber team, managers might envision starting from scratch by hiring ideal candidates based on education and experience which are shown on resumes. This assumes that the actual knowledge, skill and experience of the candidate can be gleaned from the words on the paper resume. This strategy may be ideal but probably unrealistic in both military and industry situations. In the military, commanders are selected for a position and assigned to that location with an already existing cyber team. When the commander arrives, they do not then begin hiring the ideal candidates, the team in place is already assigned to them. Similarly, in the private sector, cyber managers (who may have more flexibility with hiring and firing) come into an already existing organization and then must compete with other firms vying for the same cyber talent. Also, even if cyber leaders do have the ability to add cyber talent, this takes time. Which means that improving the cyber team outcomes might be more likely accomplished through internal team development. So, how should cyber leaders develop the team?

Cyber leaders are usually interested in developing their teams through training. This is especially true with military cyber teams who are almost always "on mission" or "training". Training will usually come in the form of individualized classroom style educational services and products where the primary outcome goal is knowledge acquisition. There is a skill identified that some personnel are lacking, and the training resolves that gap in knowledge. Some of these trainings provide certifications that are

recognized by industry by meeting a standard. Returning to the concept of individual knowledge, skills, and experience, trainings are either meant to increase the knowledge level or the skill level of the individual or both. For example, the Security+ certification is providing knowledge only. Whereas a weeklong web development software language training with a capstone project is adding to both knowledge and skill. A cyber leader sending a group of individuals to a cyber war exercise, or competition is an example of adding to the skill level only. This is what the cyber leader and management has to do: decide which trainings are most appropriate given the current environment of the team.

This is the spirit that this virtual experiment exists within. Given that a cyber leader is managing a typical cyber team, which training options should be sought after? During the focus group (detailed in chapter five), the participants reported that knowledge is easy to add, and skill is very difficult. Also, knowledge is typically not indicative of a "better" team. Skill is far more important, it is the "x" factor. Skill is also far harder to define, measure, and improve. In any event, if a mechanism for identifying and improving skill is available to leadership, how would that difference manifest within a cyber conflict? To test out this concept Cyber-FIT is used to run a virtual experiment varying team skill and adversary tier.

If skill is considered an "x" factor, then it should be a key determinant in how likely a cyber team member is to restore compromised terrain. Most cyber subject matter experts believe that experience also plays a factor because over time, being exposed to different tools and techniques, and practicing those, will inform current performance. Therefore, skill will act as a multiplicative effect on the amount of experience a cyber team member has in terms of how likely that agent is to restore compromised terrain. While tuning this version of the model, exploratory analysis showed that simply multiplying the square of skill and experience did not make a significant realistic difference in performance. To account for this, a multiplier is added, with each skill and experience combination value being weighted higher in a Fibonacci sequence. The Fibonacci sequence is frequently used in software development relative scoring systems for estimating effort and complexity in project management, so it could be a good candidate for estimating cyber operational capability as well. The defender agent restoral rate (rr) for this experiment will be based on the following equation detailed in the table below where s = skill, e = experience, and m = multiplier and adhering to the following equation.

$$rr = ((s^2 * e) * m)/5{,}000$$

e	**s**	$(\boldsymbol{s^2 * e})$	**Multiplier**	**Restoral Rate**
1	2	4	2	8/5000
2	2	8	3	24/5000

3	2	12	5	60/5000
1	3	9	8	72/5000
2	3	18	13	234/5000
3	3	27	21	567/5000

Table 33: Virtual experiment one restoral rate calculation

The purpose of this experiment is two-fold: First it will show how skill acquisition can be an extremely important strategy for cyber teams. Second, it will show that Cyber-FIT is reflecting a reasonable expectation of realism when varying team makeup and adversary complexity. For this experiment an average team will engage in cyber conflict with all six tier levels of an adversary vying for a projected cyber terrain. The cyber team will be size nine, where one agent is the team lead, four are on the network squad and four are on the host squad. The cyber team demographics will be one of three settings for the virtual experiment. The first setting will be made up of an average knowledge, skill, and experience, as reported by the cyber team survey detailed in chapter five. The second setting will increase half of the agents' skill level to three, making the average team skill level 2.5. The third setting will increase all of the agents' skill level to three, making the average skill level 3.0. In other words, this simulates swapping four average team members with expert skill team members (setting two). And then swapping the remaining average team members with expert skill team members (setting three). The details of this experiment are detailed in the table below.

Independent Variables		
IV	Variations	Range
Average team skill	5	2, 2.25, 2.5, 2.75, 3
Adversary tier	6	[1 – 6]
Control Variables		
Average team knowledge	1	2.5
Average team experience	1	2
Vulnerability growth rate	1	.001
Defender restoral rate	1	Table 31
Exploit success rate	1	1.0
Dependent Variables		

DV	Type
Compromise Time	Integer
This experiment design is 5X6X10 runs = 300 replications	

Table 34: Virtual experiment one design table

4.11.2 Virtual experiment one results and discussion

This virtual experiment, based on the underlying model assumptions and configuration, shows that adding highly skilled team members will have a large effect on cyber team performance. It also shows that an average military cyber team, with average skill, will fare well against adversaries of tier one through four, but not well against adversary tiers five and six. The following figure shows all simulation results in a scatter plot format.

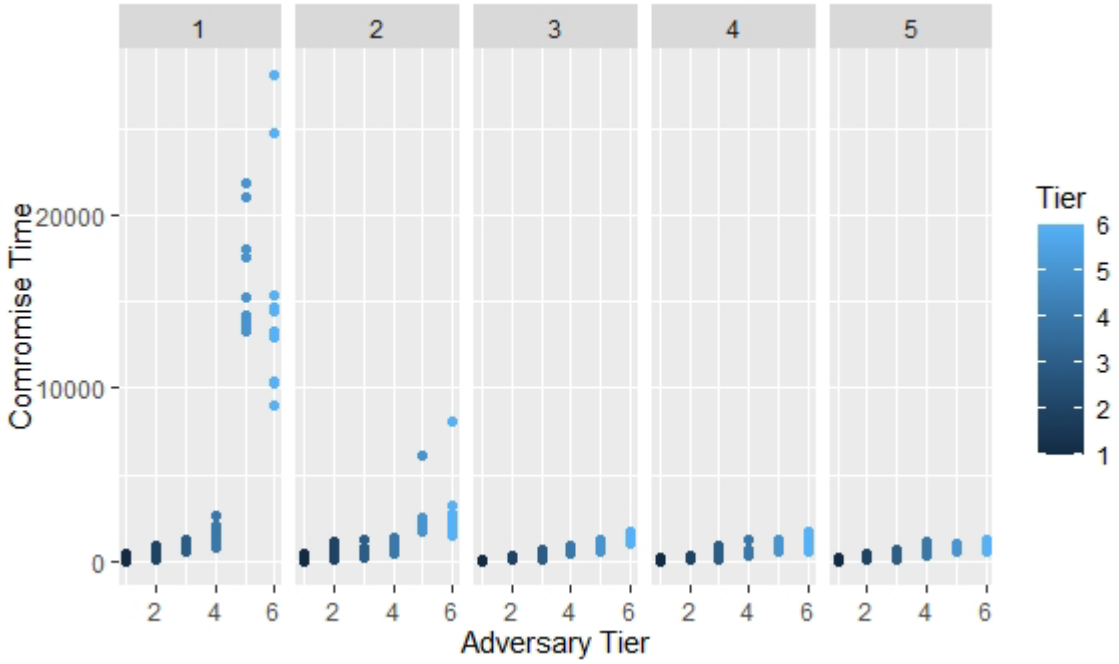


Figure 51: Scatter plot of all simulation runs across three team makeup settings

As shown, with an average skilled team, the negative effects (cyber damage) seen within the cyber terrain increases sharply as tier level is increased. The mean compromise times, standard deviations and range of simulated results is shown in the following table.

Setting	Adversary Tier	Mean Comp. Time	Standard Dev.	Range
Average	1	129.3	138.97	[0 - 473]

Skill = 2	2	493.7	244.36	[114 – 871]
	3	888.8	230.55	[587 – 1,306]
	4	1,477.7	527.14	[769 – 2,610]
	5	16,292.7	3010.52	[13,341 – 21,907]
	6	15,342.6	5949.64	[9,055 – 28,172]
Average Skill = 2.25	1	162.6	110.4	[0 – 384]
	2	436.2	316.6	[94 – 1,130]
	3	630.4	300.1	[159 – 1,288]
	4	806.8	295.6	[469 – 1,357]
	5	2,498.6	1,212.0	[1,747 – 6,085]
	6	2,735.7	1,873.5	[1,531 – 8,134]
Average skill = 2.5	1	73.8	47.94	[0 – 140]
	2	208.6	92.60	[65 – 332]
	3	374.3	208.78	[94 – 707]
	4	658.1	151.57	[400 – 929]
	5	1,011.6	206.28	[584 – 1,257]
	6	1,308.9	216.47	[975 – 1,670]
Average Skill = 2.75	1	54.3	56.7	[0 – 185]
	2	189.3	69.8	[96 – 298]
	3	418.8	246.9	[112 – 947]
	4	603.6	228.6	[302 – 1,209]
	5	885.4	229.0	[598 – 1,263]
	6	1,151.5	357.5	[604 – 1,681]
Average skill = 3	1	46.5	49.21	[0 – 164]
	2	265.2	92.90	[97 – 393]
	3	376.5	155.28	[89 – 642]
	4	570.7	251.63	[276 – 1,096]

	5	704.2	164.19	[522 – 1,017]
	6	903.9	181.37	[573 – 1,235]

Table 35: Summary of mean, standard deviation, and range compromise time simulated

The results show that a simulated tier five and six adversary will be able to control the cyber terrain and maneuver within it against an average skilled team. The cyber team will spend a significant amount of time in reactive mode, finding compromised systems and spending a significant amount of time communicating about and restoring those systems. Once a team is in reactive mode, it is hard to recover, until something significant occurs like the adversary stopping its attack, or systems being pulled, rebooted, etc.

By replacing four (half) of the average skilled agents with expert skilled agents, the setting three cyber team performs significantly better. This team is able to control the cyber terrain and quickly remediate any compromised systems they come across that have been successfully attacked by all adversaries, even tiers five and six. This means the adversary never tips the scale to the point where the team is in constant reactive mode during the remainder of the conflict. As shown, the mean compromise time from team setting one to team setting three decreases 93.8%. This result represents an approximation of reality based on what is being reported by subject matter experts on the importance of elite cyber professionals. Just adding a couple "sharp shooters" can have enormous effects. Staying with a tier five adversary, the simulation shows that moving from team setting three to team setting five results in another 30.4% decrease in compromise time. Adding more expert team members continues to have a significant effect on performance, but clearly moving from average team skill 2.0 to 2.5 will have a much bigger impact than moving from 2.5 to 3.0. Also, moving from team setting one to team setting two, where one expert skill agent replaced a medium skill agent does have a significant impact seeing average compromise time improve to from 2,498.6 to 16,292.7, a decrease of 84.7%. So, this is where the planning decision would come into play. Does the commander accept the risk of a projection of 2,498.6 with one expert or would they request one more expert to get the projection of 1,011.6? The resulting simulated data from virtual experiment one was run through a regression model with both tier and skill setting as independent variables against the log of compromise time. The regression model is shown in Figure 52 below. As shown both skill setting and tier are significant predictors of the compromise time. Tier level has a higher estimate meaning it has a greater effect on compromise time. The inflection point is clearly shown between tier levels four and five in the table above. This table also shows wide ranges of variance within simulations and virtual experiments that are clearly a combination of both stochasticity and functionality. Consider setting two where the average skill is 2.25. In this case the standard deviation of tier 5 compromise time is 1,212 for a mean of 2,498.6. By adding one more expert to the team, the mean is reduced to 1,011.6 with a standard deviation of 206.3. Another expert decreases the mean by more

than half and the standard deviation by more than eighty percent. Variance changes from both randomness and function are embedded into all simulations in this model at different levels which can be studied and improved with more validation.

```
Residuals:
    Min      1Q  Median      3Q     Max
-4.9748 -0.4248  0.0784  0.6311  1.9074

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.58084    0.19564  23.415  < 2e-16 ***
t            0.77812    0.03678  21.155  < 2e-16 ***
s           -0.38419    0.04442  -8.649  3.3e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.088 on 297 degrees of freedom
Multiple R-squared:  0.6375,    Adjusted R-squared:  0.6351
F-statistic: 261.2 on 2 and 297 DF,  p-value: < 2.2e-16
```

Figure 52: Regression model of virtual experiment 1

This virtual experiment computationally models what most would agree is actually occurring in the real world: adding enough experts to a team to make it perform at a high level. For the average team in both military and industry settings, they'll be happy to have an expert or two that can vastly improve the team. Of the thousands of teams operating in the real world, a very small number might be made up of all expert skill level members. Perhaps that most elite hacker teams, special cyber operators, and security operations centers handling high value information are made of all or mostly all expert skill. The rest are doing their best to make a marginal increase in skill level through training and practice opportunities. This virtual experiment can help them approximate their gains by doing assessments of their current strength, and what could be improved upon through training or acquisition.

4.11.3 Virtual experiment two design

The exact timing of force deployment in a military conflict is an age-old question. Sun Tzu said: “Rapidity is the essence of war: take advantage of the enemy's unreadiness, make your way by unexpected routes, and attack unguarded spots” [94]. The cyber adversary will most certainly be looking for “unexpected routes” and “unguarded spots”. Cyber team deployment is, in essence, sending guards to the cyber terrain prioritized due to mission requirements. The mission requirements are extremely dynamic and change near constantly because of changing cyberspace terrain, updated intelligence reporting,

availability of forces, changing interests, and geopolitical events to name a few. Recently, U.S. military leaders deployed cyber forces to Lithuania in response to the Russian-Ukrainian conflict due to reports of cyber attacks. Bloomberg reported: "The U.S. rushed cyber forces to Lithuania to help defend against online threats that have risen since Russia's invasion of Ukraine, an Army general said Wednesday". Notice the key term "rushed". If the forces are being rushed, it can be assumed that the mission should get started as quickly as possible, because, as Sun Tzu pointed out, the adversary is likely searching for unguarded sports. In fact, this is exactly the purpose of this cyber mission as the article goes on to say: "The so-called hunt forward missions involve cyber teams going to nations where they've been invited by partner countries, where they scan networks with the goal of building the host countries' resilience and share any new information about threats with government and private industry circles back in the U.S." [95]. This description of what the cyber forces are doing, is a specific driver of the requirements of a tool like Cyber-FIT. What will those forces do? How does the timing of their deployment affect the outcomes of the conflict in terms of availability of cyberspace terrain? Thinking through questions like this are why militaries run wargames. According to the U.S. Army Field Manual 6-0, "wargaming is a disciplined process, with rules and steps that attempt to visualize the flow of the operation, given the force's strengths and dispositions, threats, capabilities, and possible COAs, impact and requirements of civilians in the area of operations (AO), and other aspects of the situation" [96].

Cyber-FIT can be used to simulate the outcomes of importance to inform war-gaming efforts. For this virtual experiment, an aspect of a wargame that is similar to the real-world event just presented will be undertaken. In a wargame, typical "COAs" (courses of action) involve where to move forces in the terrains in play. This could be all terrains (air, land, sea, space, and cyberspace) or a subset. Imagine that the commanders, at a turn in the game, must decide whether or not to deploy cyber forces based on intelligence reporting to an area of terrain. The intelligence report provided in the wargame may indicate that sophisticated cyber adversaries are in position to attack cyber assets. The decision (course of action) could be deploy or delay. Deploy may be the correct move. Perhaps the adversary is already actively engaged, and the forces need to deploy immediately to hunt-forward and take remediating actions. Delay also may be the correct move if the intelligence reports were not accurate and the enemy is of lower sophistication and therefore easier to deal with. The delay might allow for those cyber forces to be available in a future turn where they are more useful for other campaign priorities. Thus, wargaming allows the participants to practice moves and then think through the strategic implications. A software tool that provides the cyber force deployment options and simulated results doesn't exist, and this is where Cyber-FIT can be utilized.

The motivation of a simulation, experiment, or war-game with deployment delay is a common military issue. The U.S. Army is well aware of how fast cyber defenders must

recognize and remediate attacks in order to be effective. The Army has a saying, "the golden hour", referring to how long medical troops have to get wounded soldiers attention that will increase the chances of survival. In a recent interview [97], Lieutenant General Stephen Fogarty, chief of Army Cyber Command contrasted that with cyber troop response requirements: "It's probably the golden five minutes, or that golden 20 to 30 minutes, to recognize what the adversary is putting out there and respond."

For this experiment, a cyber team will deploy to a conflict consisting of five kinetic missions connected to a tactical base infrastructure. The adversary will begin attacking at time t = 0. The cyber team will either: deploy immediately (t = 0), delay one day (t = 1,440), or delay two days (t = 2,880). Also, this experiment will include all adversary tiers, one through six. The outcome measure to consider is compromised systems, which is the number of cyber terrain in a compromised state at any given time. In real-world operations, the primary job of the cyber team is to keep that number to zero, enabling all systems to be available to kinetic forces at all times. The table below details the virtual experiment design.

Independent Variables		
IV	Variations	Range
Deployment delay time	3	0, 1, 2
Adversary tier	6	[1 – 6]
Control Variables		
Average team knowledge	1	2.5
Average team experience	1	2
Average team skill	1	2
Vulnerability growth rate	1	.001
Exploit success rate	1	1.0
Dependent Variables		
DV	Type	
Compromise Time	Integer	
This experiment design is 3X6X10 runs = 180 replications		

Table 36: Virtual experiment two design

4.11.4 Virtual experiment two results and discussion

This virtual experiment, based on the underlying model assumptions and configuration, shows that delaying the deployment of cyber forces will have an increasingly greater impact as the sophistication of the adversary is increased. This is an expected general outcome. This simulation shows that for adversary tiers one through four, the cyber team would be able to overcome the adversary quickly and return to baseline cyber terrain performance where very few compromises are being accomplished. All tiers one through four will be at baseline by day three (t = 4,3200) of the simulated conflict even if the cyber team delays its deployment by two days. Similarly, for tiers one through four, the cyber team will return the cyber terrain to baseline deployment by day two if the team delays deployment by one day. This means that whether the team delays for one day or two days, it will still take the team one day to return to baseline. This is different than the response simulated for tiers five and six. For tiers five and six, a one-day delay will take more than two days to recover from. For tiers five and six, a two-day delay could be very difficult to recover from as shown in Figures 57 and 58 below.

The utility of a simulation tool for wargaming is apparent here. The details of the game will determine how granular a simulation tool must be, for simulated outcomes. If the wargame is taking turns in a one-day time horizon, then this type of simulation would work perfectly. For example, if the participant chose to delay for two days, and the adversary ended up being tier three, and the cyber terrain was needed on day four, then then the cyber team would have been very likely to restore the terrain by the time it was needed. However, under the same circumstances, if the adversary turned out to be tier five, then the systems would be much more likely to not be available on day four (t = 5,760). The wargame might take a statistical sampling of simulation turns and then provide a key terrain cyber availability value based on the average available. The wargame might also simulate all systems in question each turn. In the former, there would have to be data processing capabilities built into the wargame software. In the latter, higher variance simulation software would have a greater affect in terms of randomization presented to the participants. In any event, this virtual experiment is a proof of concept for how Cyber-FIT, and more generally, agent-based systems should be used for higher fidelity cyber informed wargames. The following figures show all of the results of the virtual experiment.

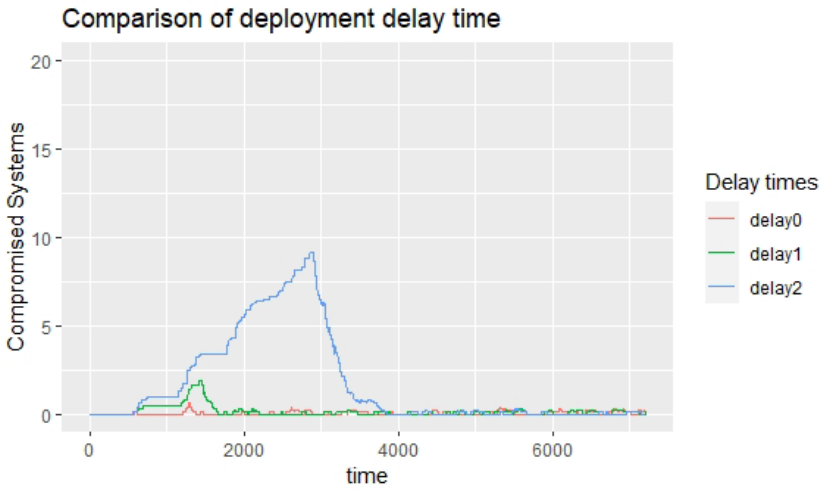


Figure 53: Tier 1 adversary cyber conflict terrain damage simulation

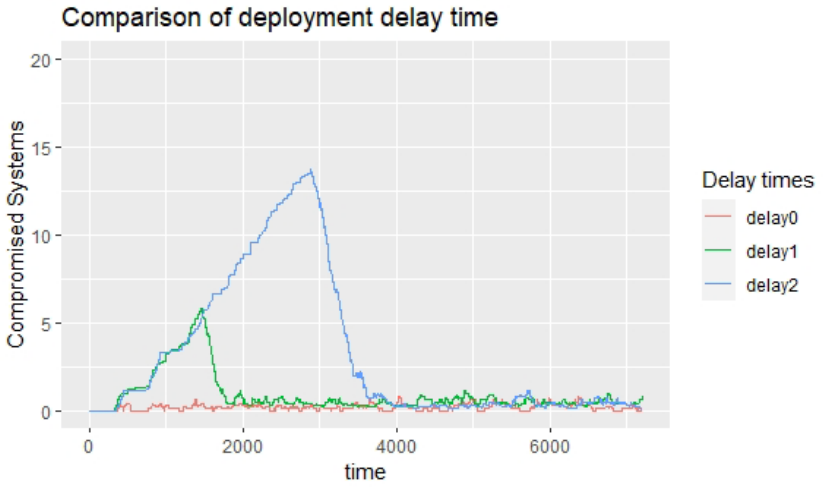


Figure 54: Tier 2 adversary cyber conflict terrain damage simulation

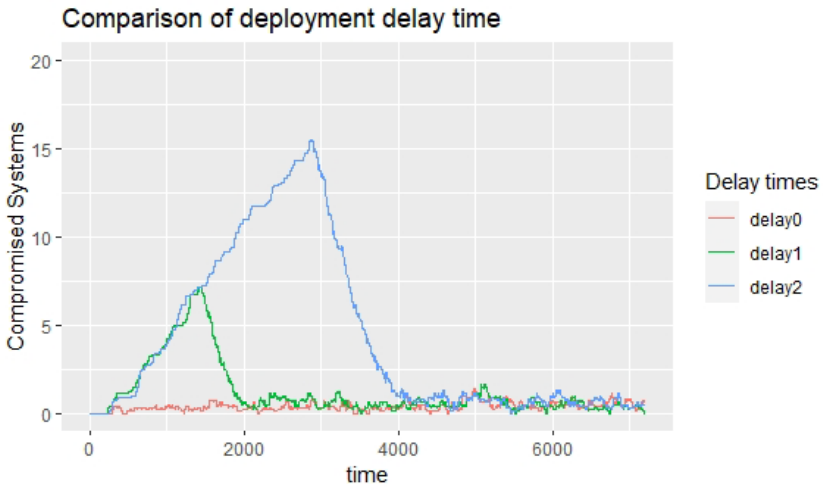


Figure 55: Tier 3 adversary cyber conflict terrain damage simulation

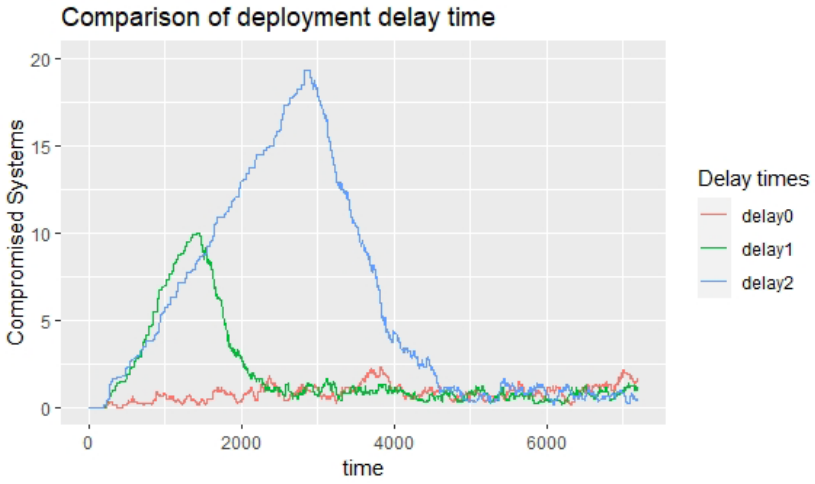


Figure 56: Tier 4 adversary cyber conflict terrain damage simulation

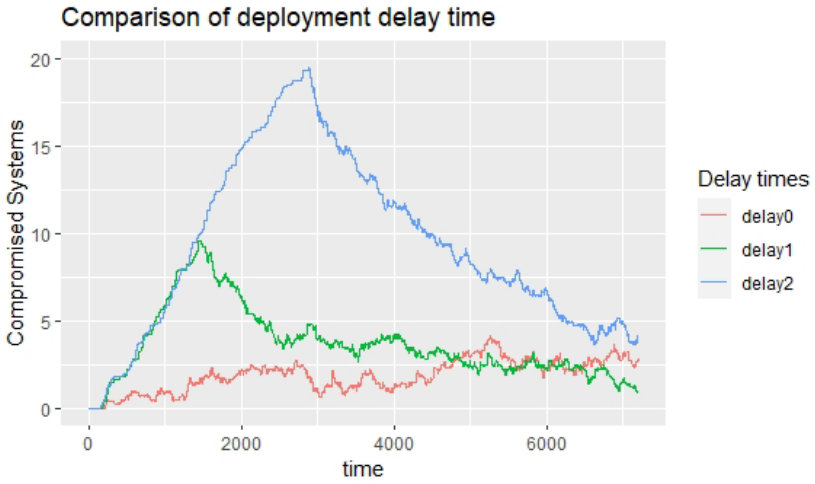


Figure 57: Tier 5 adversary cyber conflict terrain damage simulation

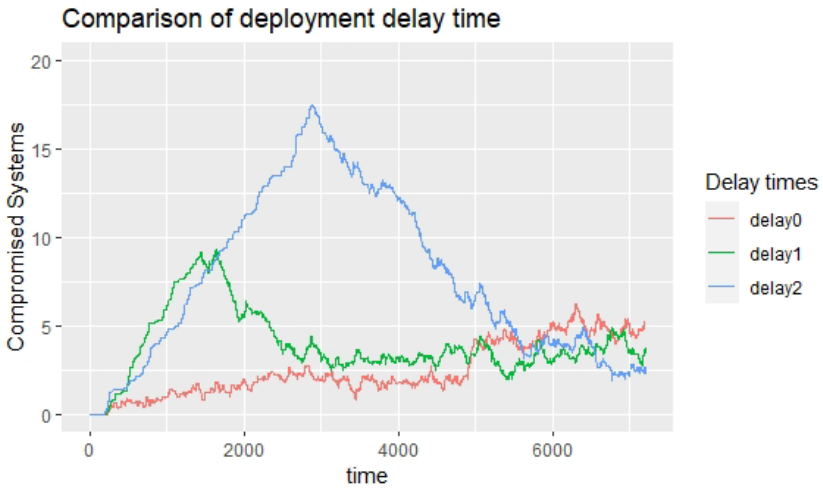


Figure 58: Tier 6 adversary cyber conflict terrain damage simulation

5 Chapter 5: Model validation

Statistician George E.P. Box said many times some version of "all models are wrong, but some are useful". One way that aphorism was expressed was this quote: "The only question of interest is 'Is the model illuminating and useful?'" [98]. This has been the guiding principle of the Cyber-FIT model development from its beginning. The real complexities of a cyber conflict are so numerous and difficult to describe, one could spend an untold amount of time trying to simply write it all down. The "truth" of what is occurring, is not of interest right now. Instead, it is an acceptable approximation of truth that can allow us to have illuminating insights and useful applications. For Cyber-FIT, this could be summarized in three categories: computational modeling, virtual experimentation, and data generation. Those applications will only be illuminating and useful if the model is approximating some level of truth. This is where model validation can assist by systematically matching parts of the model to real world phenomenon. With complex models, a strategy of validation in parts can be applied, to build the case for model validity.

North and Macal defined seven different types of validation tailored towards agent-based modeling: requirements validation, data validation, face validation, process validation, model output validation, agent validation and theory validation [99]. The next sections apply each validation methodology to Cyber-FIT, validating the model in parts.

5.1 Requirements validation

The guiding question [99] for requirements validation is: "Is the model solving the right problem"? In its earliest stages, the development of the Cyber-FIT model was searching for the problem. The "right" problem was not specifically called out, instead an abstract and simple model was put into place to generate data and potential virtual experimentation. The problem the first version was trying to solve was: "can an agent-based model simulate cyber warfare"? This turned out to be far too broad, and the problem statement was altered to read - create an agent-based model that simulates an aspect of cyber warfare that is seemingly not well understood. Cyber-FIT versions one, two, and three are essentially the process of altering and narrowing the problem to solve. Cyber-FIT version 4 is the point at which the model is solving the right problem, as will be shown through requirements validation. First, recall that the requirements were published in a series of government documents which all call for a simulation tool such as Cyber-FIT, described in chapter one. Second, quotes by senior military leaders found in news articles further validates the requirements. With enough knowledge of the problem, a requirements document is created that a software simulation tool should adhere to.

Recall the first of the United States government documents calling out the problem to solve was the Department of Defense Science Board report of 2013 titled "Resilient Military Systems and the Advanced Cyber Threat" [28]. The report was the result of a task

force examining the state of cyber security for DoD and then making a series of recommendations. Many of these recommendations were identifying an underlying core problem in the area of metrics that would tell us how to specifically measure the outcomes desired. For example, the report states "The Task Force could not find a set of metrics employed by DoD or industry that would help DoD shape its investment decisions. A qualitative comparison of resources and DoD level of effort in relation to the success rate of red teams is clear evidence of the lack of useful metrics". Not having meaningful metrics is a glaring problem. Without meaningful metrics, it is very difficult to reason about tradeoffs on decisions affecting the cyber force. Further, the report recommends the need to "increase feedback from testing, red teaming, intelligence community, and modeling and simulation as a development mechanism to build out DoD's cyber resilient force". In other words, the task force is beginning to define the requirement for a modeling and simulation tool. That tool must be able to serve as a development mechanism for designing cyber forces.

Recall the next document, published in 2015, is the Department of Defense Cyber Strategy [8]. This document had similar recommendations in terms of using modeling and simulation for assessing cyber forces. The first strategic goal of the Cyber Strategy is to "build and maintain ready forces and capabilities to conduct cyberspace operations". Several modeling and simulation points are called out as requirements supporting the strategic goal. The first is to "establish an enterprise-wide cyber modeling and simulation capability" and "develop the data schema, databases, algorithms, and modeling and simulation capabilities necessary to assess the effectiveness of cyber operations." This means that a software must be designed (most likely an object-oriented software) that has integrated functions programmatically enforcing algorithms which ingest data, make changes, and then output data in schemas. The output data can be stored in databases or file systems. Next, the strategy calls for a need to "assess cyber mission force capacity" to "achieve its mission objectives when confronted with multiple contingencies". This language is defining a requirement that the software output should be associated with mission metrics. Also, it must be able to handle multiple scenarios of varying situations that the cyber force could be confronted with. Lastly, in this document, the "Joint Staff…will propose, collect, analyze, and report a set of appropriate metrics … to measure the operational capacity of the CMF". This language is defining requirements for the modeling software around how the output data should look. It should either define the metrics of interest or define the output data such that post-processing software would be able to show desired metrics. Ideally, a multitude of data would be available so any number of software applications could ingest and utilize the data in novel ways.

Recall the next document, published in 2019, is the White House Executive Order on America's Cybersecurity Workforce [29]. The order states: "The Secretary of Homeland Security, in consultation with the Secretary of Defense, the Director of the

Office of Science and Technology Policy, the Director of OMB, and the heads of other appropriate agencies, shall develop a plan for an annual cybersecurity competition (President's Cup Cybersecurity Competition) for Federal civilian and military employees. The goal of the competition shall be to identify, challenge, and reward the United States Government's best cybersecurity practitioners and teams across offensive and defensive cybersecurity disciplines". The competition is now very popular and has been held several times, with some lawmakers proposing bills to codify it as a yearly budget item [100]. The competition challenges participants with cyber skill games where they gain points and move through rounds ultimately vying for the championship trophy. This means that there is a scoring aperture that defines who has the best skill and differentiates individual and team, adding another requirement to the simulation software.

The last document of interest is the United States Air Force Broad Agency Announcement #FA8650-20-S-6099. The primary purpose of the announcement is to solicit research proposals that propose novel ways to understand learning and performance within Air Force personnel training systems [101]. Of most interest is the language around defining performance, and how knowledge, skills, and experience interact to influence performance. The announcement is defining requirements for simulation software such as: "having very specific and valid knowledge, skill, experience, and performance (K, S, E, P) information on what airmen in various roles are expected to be able to know and do and how they are expected to operate and perform". This means that agents should have defined class variables tracking all individual traits (K, S, E) and then defining performance metrics that result from these traits. The announcement also requests: "methods to define and quantify what is meant by terms such as 'proficient,' 'mission ready' and 'effectively and efficiently operating'". This adds another requirement for the software which ties mission data and objectives to both the team activities and the team makeup.

Software development requirements writing typically begins with customer, or external user desires, and then moves to the internal development necessary to set up the appropriate objects and data structures to deliver the higher-level capabilities. For Cyber-FIT version 4, this is described in the table below. The Observer class of Cyber-FIT software is an abstraction of what the components are doing collectively to meet the requirements called out by the four government documents. Following the Observer class are the main component classes making up the agents, interactions, and mission objects that must operate independently to achieve the collective system functionality. The tables below are written in the Agile Development User Story method, by software class, completing the requirements validation exercise for Cyber-FIT.

Observer Class		
No.	User Story	Done

1	As a model user, I want to simulate effects of cyber forces as a development mechanism in building a resilient cyber force	X
2	As a model user, I want to have access to a well documented software that defines algorithms, models, and database objects for cyber modeling and simulation	X
3	As a model user, I want to simulate different cyber mission types against multiple types of adversaries	
4	As a model user, I want to export the simulation data in various industry standard formats such as JSON, csv, XML, etc	X
6	As a model user, I want the formulas which describe mission success embedded into the software, or the data output in a way that post-processing software can define the formulas	X
7	As a model user, I want the data that is output to be in a format that can define individual metrics, team metrics, or a combination of both	X
8	As a model user, I want a mechanism in the form of configurations that can incorporate theoretical models such as cyber situation awareness and organization theory	

Table 37: Observer class requirements in Agile user story format

Terrain Class		
No.	User Story	Done
1	As a researcher, I want to model the effect of computing systems incurring vulnerabilities,	X
2	As a researcher, I want to model different behaviors, as a result of external stimuli, of different types of computing systems in terms of network architecture, routing, and information sharing	
3	As a researcher, I want the computing systems within the model to track vulnerability level, malicious code present, and compromise status	X
4	As a researcher, I want to track computing systems to missions for mission level analysis	X

Table 38: Terrain class requirements in Agile user story format

Defender Class		
No.	User Story	Done
1	As a researcher, I want to differentiate the team members by squad for modeling of differentiable sub-element behaviors	X
2	As a researcher, I to model the basic operations of a cyber defender in terms of survey, securing, and protecting terrain	X
3	As a researcher, I want to change behaviors of individual agents based on frameworks such as NICE [102], along with emerging policy and doctrine	X
4	As a researcher, I want individual team members to track demographic data based on typical military and industry reporting requirements	
5	As a researcher, I want to model cyber incident response procedures	
6	As a researcher, I want the cyber team members to interact with each other in the form of information sharing, reporting, and directives	X
7	As a researcher, I want the cyber team members to interact with computing systems differentiated based on the purpose of the cyber operations they are working	X
8	As a researcher, I want the cyber team members to behave differently based on their level of knowledge, skill, and experience	X
9	As a researcher, I want the cyber team members to have functionality representing a cognitive model of the situation that changes over time	

Table 39: Defender class requirements in Agile user story format

Attacker Class		
No.	User Story	Done
1	As a researcher, I want to model the attacking agent behavior moving through steps such as the cyber kill chain, or other similar intrusion chain methodologies	X
2	As a researcher I want to control behavior within the steps or phases of the	

	intrusion/kill chain methodologies	
3	As a researcher, I want the attackers to have varying complexities leading to differential behavior	X
4	As a researcher, I want the attackers to track summary statistics about success and failure per attack attempt	X
5	As a researcher, I want the attackers to have to be dependent on vulnerabilities existing on terrain they are attacking for success criteria	X
6	As a researcher, I want some high-level attackers to be able to exploit zero-day vulnerabilities	X

Table 40: Attacker class requirements in Agile user story format

Friendly Class		
No.	User Story	Done
1	As a researcher, I want the non-cyber agents to be modelled as using the computer systems for the purpose of their missions, tracked by mission	X
2	As a researcher, I want the friendly agents to request information and track whether or not the information was received, how quickly, and of what quality	X
3	As a researcher, I want the friendly agents to share information that is received with other team members	

Table 41: Friendly class requirements in Agile user story format

Interaction Class		
No.	User Story	Done
1	As a researcher, I want interactions to be tracked by which classes are being connected, with differentiating behavior basis	X
2	As a researcher, I want to model interactions by type which determines how long they persist	

Table 42: Interaction class requirements in Agile user story format

5.2 Data validation

The guiding question [99] for data validation is: "Have the data used in the model been validated"? This refers to the input data the model ingests along with data that controls agent behaviors, typically called control variables. The data used for Cyber-FIT come from a mixture of sources such as official government websites, sampling, interviews with experts, policy-based literature, and empirical studies. With any novel model, there will be some aspect that doesn't have data or literature backing it, which is typically why it is new and of interest to emerging research. For Cyber-FIT there are several examples of this and can be outlined with respect to the three most important agent classes: terrain, defender, and attacker. The table below details the most pressing model behaviors per agent class that would make the system as a whole more realistic if data were available.

Class	Behavior	Data needed
Terrain	Connecting computing systems, enforcing networking rules, reading and writing information	Routing protocols, typical network architectures, empirical network data
Defender	Operational behavior detailing type, resources needed, frequency, and reporting	Team makeup demographic data, self-assessment, types of operations, typical communications
Attacker	Cyber attack campaigns based on motivation and group affiliations	Attack patterns affiliated with organization and artifacts associated with specific techniques

Table 43: Data identified as necessary for a realistic simulation of cyber engagement

When building a model from scratch, designers must be selective in what behaviors to add. Too many additions at once introduces the risk of too much complexity too quickly, resulting in outcome variables which are hard to disentangle from behaviors. It's typically advised to add data and behaviors one at a time, so that those behaviors can be validated along the way. This is the tension between transparency and veridicality which can be analogous to simplicity and complexity [103]. Cyber-FIT was built using a spiral development methodology, each version adding a minimum amount of complexity, while adding research functionality as shown in the figure below.

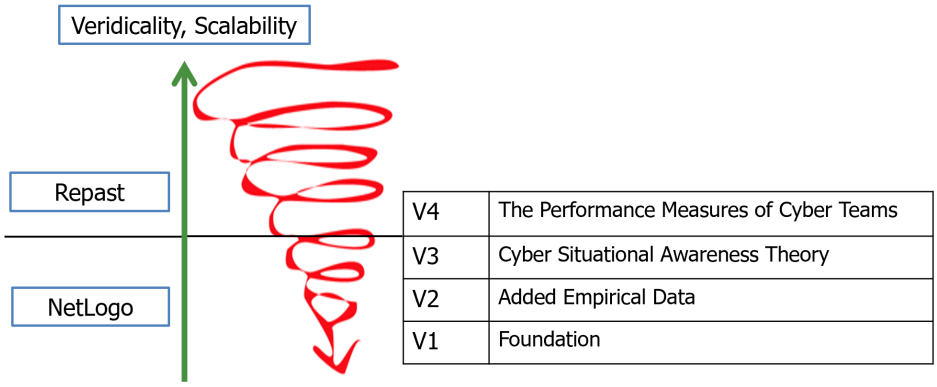


Figure 59: Spiral development strategy by version

As previously described in chapter two, the original model defender agents only looked for vulnerable and compromised terrain agents and attempted to fix on the spot. If they were unsuccessful, they moved on to others. In Cyber-FIT version 4, a more complex defender agent behavior was designed. This was in the form of 1) keeping track of the known system vulnerabilities, 2) continuing to attempt to restore known compromised machines at a restoral work selection rate, 3) communicating with other team members about known vulnerabilities and compromises, 4) selecting terrains to interact with based on squad assignment, and 5) selecting operations from a list. This level of veridicality was a sufficient enough change to observe outcome variables in the form of virtual experiments, sensitivity analysis, and validation efforts. From the table above, this is the defender agent behavior that would be updated. Future versions will address more complexity in terms of terrain and attacker behaviors.

While considering the behavior changes for defender agents, many questions come up such as: How many team members are on a typical team? What is their makeup of knowledge, skill, and experience? How often should team members send information to each other? How long should operations take? How much time transpires between operations? Does it take longer to conduct operations based on knowledge, skill or experience? The list of questions can go on and on. With the primary design change in this version being the behaviors that could be influenced by knowledge, skill, and experience, a survey could help answer some of these questions. A Qualtrics survey was designed to collect data for this purpose. The questions were presented in three parts: demographics, interactions, and performance. The sixteen-question survey was taken anonymously by a random assortment of cyber security professionals advertised through several cyber security email distribution lists.

The first questions in the demographics category asked the respondents to report on information regarding experience type, experience length, self-assessment of skill, and level of education. The answers to these questions provided data with which an approximation of the typical team can be made. The typical team has 6 – 10 personnel,

has a wide variance of experience, and is highly educated. The next questions in the interactions category asked about the frequency of types of operations and communications during normal operations and during incident operations. The data shows that cyber team members conduct a variety of operations and during incidents, the frequency of communications and interactions goes up. Finally, the questions in the performance category asked questions about how the respondents assessed their own performance, and how their performance was assessed by leadership and management. The data show that cyber team members don't typically understand how well their team is performing, and there is a wide variety of ways that teams are assessed. All survey data and summary statistics is provided in Appendix A.

With the completion of the cyber team survey, the data validation goal for this version of Cyber-FIT was reached. In version 4, there are several sources of validated input data affecting model behaviors coded in the software. This adds to the overall validity of the system processes and the realistic behaviors and output data, which is described in the next section in the form of face validation. The table below details all the input data and control variable data that is based on either empirical data or system behavior from literature.

Input Name	Description	Data Source
Terrain Vulnerability Growth Rate	The terrain vulnerability growth rate (VGR) is based on the MITRE CVE database which tracks all known software vulnerabilities per operating system type and version. The VGR can be controlled by OS type, patch level, environment, or timing within the mission.	MITRE Common Vulnerabilities Enumeration database [34]
Defender Knowledge	Defender knowledge level is a quantification of the sum total knowledge acquired over the individual's cyber security career which can be made up of formal education and certifications.	Cyber Team Survey
Defender Skill	Defender skill level is a quantification of the inherent level of skill the individual possesses. This agent trait is most difficult to quantify due to its nebulous nature.	Cyber Team Survey
Defender Experience	Defender experience is the easiest trait to quantify as it is simply the amount of time the individual has spent working in the cyber security industry	Cyber Team Survey

Defender Cyber Operations	The defender agents, when conducting normal operations pull from a list of operation types based on the CISA NICE Cyber Security Framework	CISA cyber operation types [102]
Defender Interaction Rate	The defender agents will choose to interact with other defender agents or terrain agents and this will be increased when cyber incidents are occurring modeling real world bustiness	Cyber Team Survey
Attacker Kill Chain Phase Time	The amount of time an attacker spends in each of the cyber kill chain phases	Empirical Data [104]
Attacker Zero-Day Development	The chances that a tier six attacker agent can develop a zero-day attack	Empirical Data [105]
Attacker Tier Level	The tier level of the attacker agent from one to six based on Defense Science Board Report	Defense Science Board Report [28]
Mission Terrain Configuration	The number of networking devices, servers, and host terrain agents per mission	Empirical Data [106]
Mission Cyber Operations	The cyber mission to conduct which are one of three types: survey, secure, and protect	Gaining Cyber Dominance Technical Report [56]

Table 44: Input data affecting behaviors and control variables

5.3 Face validation

There are two guiding questions [99] for face validation. The first is: "when looked at in a systematic way, do the assumptions upon which the model is based seem plausible"? The second is: "do the model results look right"? The face validation of Cyber-FIT was accomplished by holding a focus group of three experienced cyber security subject matter experts. All three participants have more than twenty years of cyber security experience. One has active-duty military experience only, another has active-duty military experience, is retired, and now has six years of industry experience, and the third has over twenty years of industry experience. This mixture was sought so that active-duty only, active-duty/industry mixture, and industry only perspectives would be included. Cyber-FIT is a military style designed software simulation tool, but the concepts are general enough that

someone without military experience would still understand the underlying cyber security concepts. In fact, most industry security operations centers operate in similar fashions to military cyber protection teams, so the carryover is apparent.

The focus group was held in January 2022 over zoom for one hour. There were two parts to the focus group. In part 1, the previously described cyber team survey was reviewed in order to face validate the responses. All three individuals validated that their experience was in line with the survey responses. Two questions were of most interest to the focus group. The first was the question where 19 out of 20 respondents said that interactions amongst team members increases during cyber incidents. According to the focus group there are three important dimensions of the burstiness in communications and activity associated wit this team behavior, all based on stress. The first is stress around environment familiarity. Stress will manifest itself in different ways most usually associated with how well the team knows the environment (the cyber terrain). Teams that are unfamiliar with the computer network they are protecting will have a much higher amount of stress. This will lead to looking for things in a myriad of places because they might not understand exactly what certain security tools are reporting on, or the details of the configuration. Teams with more experience and knowledge of the environment will be able to dial into the tools that are most useful for that specific problem they are seeing. Cyber-FIT does model uncertainty with the agents, some percentage of the time, doing nothing, because of confusion. Also, Cyber-FIT will increase the interactions between cyber team machines and network machines, increasing the computer-computer connections. The experts agreed this was a potentially useful behavior and could be extended in many ways based on team member variables. The differences that the experts know happen in real operations could be experimented with. Second, stress will increase as time goes on if the problem is not identified. Teams would typically become hastier in their searches over time, and the searches (connections to machines and observing of dashboards) would go deeper into the network. Also, the interactions amongst team members would increase and would be apparent through a variety of tools. Finally, stress is highly dependent on reporting requirements. Most security operation center, and certainly all military cyber teams, will have specific reporting instructions that must be followed based on the severity of the incident. The higher the severity, the higher the stress. If there is a report due every hour, with updated details about what is being done and what has been found so far, then that will drive the activities of the team. The higher up the reporting chain in the organization, the higher the stress and the higher number of managers involved. All three of these stress responses could be modeled, simulated, and experimented with in Cyber-FIT.

The other cyber team survey question of particular interest to the focus group was about perceptions of how well cyber teams perform. The question was “On cyber teams you’ve been on, do you typically have a good understanding of how well the team is

performing?". Six respondents said no, two respondents said unsure, and eleven said yes. Overall, 42% of respondents either didn't know or were unsure. If this is representative of cyber teams at large, this is an enormous number of cyber security professionals not understanding if their operations are positively affecting the organization. The focus group participants were not surprised by this result and thought it tracked well with their experience. As experienced cyber security professionals, senior among their peers and typically in leadership positions, they did have a sense of how their teams perform, but it would be hard to quantify, which is one of the primary drivers of Cyber-FIT development. The focus group brought up several reasons for this overall misunderstanding. First is attrition amongst the information technology professionals in both military and industry environments. Measuring performance successfully is a long process of baselining, setting improvement targets and then re-assessing. All of those activities are measuring the skills of the people involved. If the team experiences turnover, then the people are different, and some sense of measurement is disrupted. Another issue similar to attrition is the changing technological environment that the team works in. If new cyber tools are introduced, or there are major changes to the network being protected, this changes how the team works and throws off previous measurements of performance. This means that the best measurement methodologies should be tool and environment agnostic, instead focusing on the team behaviors and processes.

The focus group gave feedback about how teams are typically assessed. "Purple" teams are an industry standard where the team will exercise how an attacker (red) and defender (blue) might engage in the operational environment. This can be a tabletop exercise where documentation is examined, and a leader works through a set of questions. Another way that teams are frequently assessed is through external audits like a consulting company testing the team, or a penetration testing team attempting to hack into the network. The focus group also identified cyber competitions and exercises as a way that teams are currently assessed. Competitions can show how teams stack up against other teams. The problem is that the competitions are almost never a realistic matching to what the team's do in normal operations.

Next, the focus group discussed ways that team performance could be assessed given no financial or resource constraints. They all agreed that the ultimate mechanism would be a high-fidelity cyber range in a virtual environment where any possible cyber incident that might come their way could be simulated, appearing just like how it would manifest in their own network. Essentially, a practice field that is a replication of their real field, just like sports teams practice on. There is existing literature [107] [108] on this line of research and clearly an active need for military applications. If the cyber range is up and operational, the organization can run two teams through the same scenario and see who does better. Since the vast majority of organizations cannot afford to create a realistic practice range, the next best option is using the organizational data already present and

define business metrics that are tailored to the organization. This is hard in practice, but good cyber leadership is able to make things like this happen. For example, most cyber teams use an incident tracking system of some sort with "cases" that occur. The case data is a good source of information for what was discovered, how quickly, who worked on it, was the appropriate attributions made, etc., etc. By creating a standardized reporting process, analytics can be developed showing how well the team is performing. In a similar mindset, data about what the users are doing on the network can also be used to approximate team performance. If users need to access certain assets, then how well the information technology is providing that access is a performance measure. Server logs, internet traffic, and database logs can provide that information.

The focus group then received a demonstration of Cyber-FIT version four with a realistic simulation and walked through the output measures (cyber team performance measures). The focus group agreed that time to compromise (from attacker perspective) is one of the most used within training and exercises in controlled environments. Cyber teams will defend a network and the attacker will try to compromise machines, with the best teams able to maximize time to compromise. The other that are most prevalent, according to the focus group are: time to detect and time to restore. Finding and fixing problems is the primary purpose of the cyber defense team. Finally, compromise time is also one of the most important metrics because it is essentially combining time to detect and time to restore. All of the other measures make sense from a cyber leadership perspective but aren't currently being tracked because of the difficulty in gaining the relevant data.

The final portion of the focus group was discussing applicability of Cyber-FIT to real world problems. Each member of the focus group was able to provide a different use case that would be of interest to parties in positions of cyber leadership. The first would be to use it for war-gaming as course of action (COA) analysis. COA analysis is used at high levels of military analysis and there is little in the way of war-gaming for cyber currently available. Secondly, the model could be used for policy analysis. Many cyber policies use language that is vague. Running a Cyber-FIT simulation with the definitions laid out by NIST and the DoD would provide a computational analysis of what the policy is prescribing in the form of cyber work roles. Last, the model can be used for virtual experimentation when it comes to assisting in decision analysis. How should a cyber team be trained and when should it deploy? This is done by altering configurations of input and control variables and observing differences in outcomes. The virtual experiments completed in chapter four were run as a result of this request.

Overall, the focus group was extremely positive in the usefulness of the Cyber-FIT framework. There is certainly a gap in the science that this model is addressing due to the fact that none of the focus group members have ever come across a tool that is addressing an extremely clear need. The two questions of face validation are affirmed to be positive.

5.4 Process and agent validation

North and Macal, when listing [99] out the validation types for agent-based modeling list "process validation" as a different type than "agent validation". The guiding research question for process validation is: "Do the steps in the model and the internal flows of what is being modeled correspond to the real-world process"? Separately, the guiding research question for agent validation is: "Do agent behaviors and interaction mechanisms correspond to agents in the real world"? For Cyber-FIT, it would be too difficult to separate those two questions. The agent behaviors and interaction mechanisms are the internal flows. For other agent-based models, that depend on more business rules, or already existing activities and processes that are independent of the agent behaviors, this separation would make sense. But for Cyber-FIT both questions will be addressed with the same analysis.

To begin with process validation, consider the workings of all the agents together in the model. The figure below shows a screenshot of the Cyber-FIT version four user interface along with pictorial representations of the agent types. As shown, there are four agents working together: terrain agents, friendly agents, defender agents, and attacker agents. Each agent process will be described next.

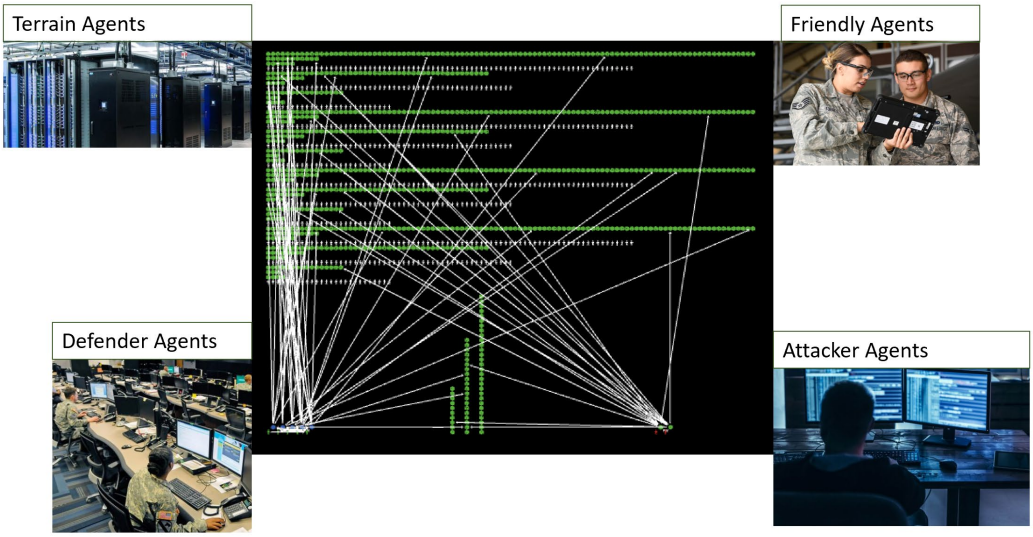


Figure 60: User interface of Cyber-FIT version 4 with visual aids representing agents

5.4.1 Terrain agent process

Each time step all terrain agents stochastically generate new vulnerabilities that are added to its vulnerability array variable. This is based on the terrain vulnerability rate control variable and vulnerability generation method. Next terrain status values are

updated based on if other interactions have changed the terrain's status from one possible state to working, degraded, or compromised. Finally, terrain statistics are updated collecting temporal data on vulnerabilities and terrain status. The terrain agent step algorithm is in the next table below.

Algorithm: Terrain Agent Step
1: **if** (generate_vulnerabilities == true)
2: **then** add vulnerability ID number to *vulnerability[]* array
3: Update *terrain_status* values
4: Generate terrain_statistic values

Table 45: Terrain agent step algorithm

5.4.2 Defender agent process

Each time step all defender agents either complete restoral operation, continue their current operation, or get a new operation. During restoral operations the defender agent will connect to its workstation and then connect to a known compromised terrain agent to attempt to restore it. If the defender agent is not aware of any compromised terrain, it will either continue its current operation, or select a new operation to conduct. Finally, the defender agent updates values associated with team performance and mission information. The defender agent step algorithm is in the next table below.

Algorithm: Defender Agent Step
1: **if** (compromised_terrain == true)
2: **then** restoral operations AND message_team_lead
3: **else if** (*operation_complete* == false)
4: **then** continue_operation AND message_team
5: **else** get_operation
6: Update *cyber_mission* values
7: Update situation_awareness values
8: Update *performance* values

Table 46: Defender agent step algorithm

5.4.3 Friendly agent process

Each time step all friendly agents stochastically interact with cyber terrain agents to read information from that cyber terrain. The terrain agent will send back information if it is normally operating (not compromised) at a speed based on the vulnerability level of the cyber terrain agents associated with that mission. This is simulating the primary usage of the computer network: reading information necessary for doing their job. The friendly agent step algorithm is in the next table below.

Algorithm: Friendly Agent Step
1: **if** (get_information == true)
2: **then** read_mission_terrain_agent
3: Update *cyber_mission* values

Table 47: Friendly agent step algorithm

5.4.4 Attacker agent process

Each time step all attacker agents work through the cyber kill chain which is: reconnaissance, weaponization, delivery, exploitation, command and control, and actions on objectives. Each phase has stochastic elements regarding time spent in the phase along with behavior and success criteria. The attacker agents will update performance values along the way in certain phases. The attacker agent step algorithm is in the next table below.

Algorithm: Attacker Agent Step
1: switch (phase)
2: case 0: **if** (*phase_complete* == false)
initialize_attack_resources
else set phase = 1
3: case 1: **if** (*phase_complete* == false)
read_terrain_vulnerabilities
else set *phase* = 2
4: case 2: **if** (*phase_complete* == false)
weaponize_attacks

		if (weaponize_fails) set phase = 1 || 0
	else set *phase* = 3	
5:	case 3: **if** (*phase_complete* == false)	
		deliver_payload
	else set *phase* = 4	
6:	case 4: **if** (*phase_complete* == false)	
		if (attack_success) set phase = 5
	else set *phase* = 0	
7:	case 5: **if** (*phase_complete* == false)	
		command_and_control
	else set *phase* = 6	
8:	case 6: **if** (*phase_complete* == false)	
		actions_on_objectives
	else set *phase* = 7	
9:	Update attacker_statistics values	

Table 48: Attacker agent step algorithm

5.4.5 Process and agent validation summary

The process and agent validation of Cyber-FIT was completed first through many conversations with subject matter experts, cyber military personnel, and other researchers. Then it was finalized with the focus group described in the previous section. Ultimately, each agent is following internal flows and processes that map to real world behaviors. One way this has been designed, is by imagining any behavior that could possibly be added to the model and ensuring that there is a place for that behavior. If there is, then the underlying mechanism is corresponding to real world behavior, at a basic level. For example, if a use case emerged to study how cyber operation error rates would affect the team performance, there is a place in the continue_operation() method in the table above to add an error rate that would affect the cyber operational behavior of the defender agent.

5.5 Model output validation

The guiding question [99] for model output validation is: "if the real-world system is available for study, do the model outputs match the outputs of the real-world system?" This validation type is traditionally what people think about when they think of

"validation" in a general sense. Does the simulation match reality? Obviously, this is very difficult. Difficulties arise for many reasons including scope of simulation, complexity of the model, difficulty with noise and perturbations, and availability of empirical data. In many instances, the ultimate goal of a simulation model is to go from idea to grounded theory. An example is using the Construct simulation tool to simulate organizational behavior, validated by empirical communication data resulting in a grounded theory of referential data knowledge transfer [109]. Cyber-FIT could potentially lead to grounded theory on a cyber team performance. Cyber-FIT model output was validated twice over the course of this work. The first time was using the empirically observed cyber intrusion chain behavior described in chapter two. The second is a comparison of computer interaction data of a Cyber-FIT simulation and empirical network data.

5.5.1 Model output validation of cyber kill chain

In chapter two, the second version of Cyber-FIT was described in detail explaining the motivation to add empirical data to the model to increase realism and complexity. At that time, the goal was not to add more complex attacker behavior by finding an empirical data set. Instead, it was to first find a data set and then figure out if that data set was applicable to Cyber-FIT. Considerable time was spent pouring through publicly available data sets and even some that were restricted use. A serendipitous moment occurred when I was attending a conference and met a group of students at a poster session (plug for poster sessions!). They described a force-on-force cyber exercise where they observed all of the attacker team behaviors and annotated the details of their behaviors. This was a perfect data set to incorporate into Cyber-FIT and could extend the existing version one attacker behavior. This table below was previously presented in chapter two.

Phase	Empirically Observed Time	Average Time Simulated	Range Simulated
1	75	79.08	[61 - 141]
2	50	55.95	[36 – 156]
3	20	144.58	[10 – 2,281]
4	35	48.37	[5 – 589]
5	20	24.17	[6 – 105]
6	85	88.88	[71 – 138]

Table 49: Cyber-FIT version two virtual experiment simulating empirically observed data

There are many ways to conduct the analysis of output for validation. It could simply be observing specific values, on the low end, to complex statistical analysis on the high end. The decision usually centers around: what is the purpose of this analysis and what is good enough? Since the empirical data observed was one sample and not a distribution that could be averaged, there's no need for advanced statistical analysis. Instead, the simulation should be able to simulate a variety of outcomes where the empirically observed data is found within the range of simulations. The table above shows that this was the case. More importantly, the cyber kill chain should show different ranges of values due to what must happen to move from one phase to the next. This work showed, in a novel way, how to use agent-based modeling to accomplish that. Attacker agents must meet success criteria at different phases based on varying rulesets across environments and configurations. This rulesets and settings can be experimented with in many other ways. Version two successful incorporated and validated the output of the model with empirically observed human actor behavior.

5.5.2 Model output validation of computer network interaction data

In previous work, a cyber situation awareness dashboard was improved with network science data [110]. That research was asking the following questions: "*does binning data, and then calculating graph level measures, provide a more granular picture of the network so that anomaly detection is easier to accomplish? If so, can we create "normally operating" network science-based signatures? How can organizations use these insights, incorporating their known patterns of life, to* achieve enhanced cyber situational awareness?" In summary, that research was able to show three key findings. First, binning the data did result in different distributions of network measures. Second, the network data showed strong signs of periodicity. Third, patterns of life incorporated into dynamic network analysis improved cyber situation awareness.

That work went through a systematic three step process to gather and analyze the data. The first step retrieves flow records into four bins using criteria to separate human and autonomic in and out traffic. These records are exported in comma-separated-value format. Step two imports the saved files into the CASOS tool ORA and converts them to DyNetML files, allowing for network data inspection. Step three conducts a dynamic network analysis on the four datasets measuring various network measures on an hourly basis. Autonomic in traffic is traffic flowing into the network and likely generated by computer software (no human engagement). Analysis of the autonomic in traffic, showed that the average density over the entire data set was .0002 and average network centralization total-degree was .0004. These measures were found on NetFlow covering approximately 25,000 nodes (computer hosts). The table below shows the binned network measures results from the study.

NetFlow Type	Avg. Density	Std. Dev	Avg. Centralization, Total Degree	Std. Dev
Autonomic In	0.000243	0.000081	0.000405	0.000621
Autonomic Out	0.000189	0.000080	0.000981	0.000265
Human In	0.000096	0.000034	0.000872	0.000321
Human Out	0.000130	0.000049	0.001075	0.000299

Table 50: Binned NetFlow network measures from empirical data study

So, will Cyber-FIT output similar network measures? This will be an excellent test of the framework. To test this Cyber-FIT is setup to support 20 kinetic missions consisting of 2,500 computer nodes. For this simulation, both the cyber team (defender agents) and the adversary (attacker agents) are turned off. This simulates removal of the human traffic within the network. Also, since Cyber-FIT only simulates the internal network traffic, this is akin to the bin of in flow traffic only. Essentially, this simulation is setup to only track the autonomic inflow, like the empirical data set it will be compared to. The simulation is run for five simulated days (7,200 ticks) and the terrain agent to terrain agent interactions are collected. This data is ingested into ORA and analyzed as a dynamic network over five simulated days. Next, a sample of the NetFlow data (one hour of each day of the full set) is ingested into ORA and key framed by day (similar to the Cyber-FIT simulation data) in order to do a side-by-side comparison. This sampling of empirical data includes all bins, but as noted in the table above the different bins don't display huge differences, they are all on the scale of 10^{-3}. We are hoping to see that Cyber-FIT can output on the same scale. The figure below shows the results of a dynamic network analysis using ORA for both the empirical data (on the left) and the simulation data (on the right).

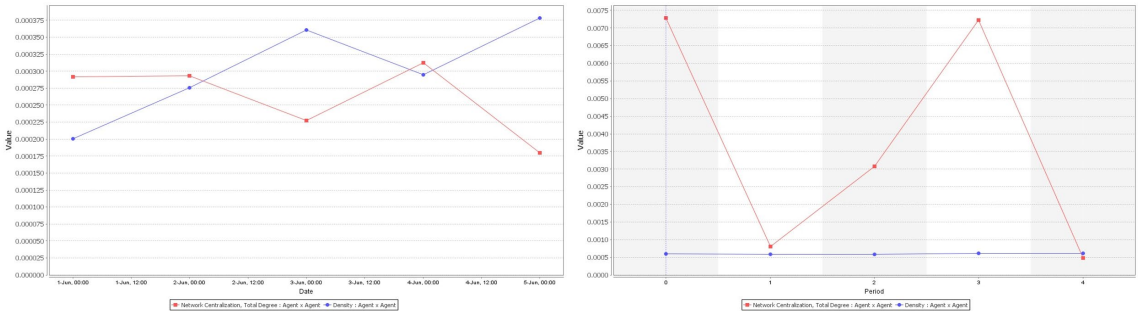


Figure 61: Sample of empirical NetFlow data versus Cyber-FIT simulated NetFlow data

The results show that Cyber-FIT is able to simulate similarly scaled network measures. On the left, the empirical data sample, over five days, results in a range of density values of [0.000200 – 0.000378]. The empirical data results in a range of network centralization total degree values of [0.00018 – 0.000313]. The simulated data results in a

range of density values of [0.000590 – 0.000611] and a range of network centralization total degree values of [0.000473 – 0.007282]. In summary, the model is outputting network data that is measured at the graph level on the same order as empirical data. This shows that the framework overall is doing its primary job of being a test bed with which experimentation can lead to very realistic simulation outcomes. The simulated data shows much higher variance in network centralization total degree than density, as expected. The software is simulating connections amongst networked computers at a low rate and this means that over time approximately the same number of connections will occur during time periods. However, the randomization of the structure of the networked computers will be different in different time periods. This is due to the current version of Cyber-FIT not enforcing routing protocols. Any terrain agent can connect to any other terrain agent. In the empirical data this is different. The empirical data shows low variance in network centralization total degree because the structure of the network will be similar between time periods because routing protocols are enforced.

5.6 Theory validation

The guiding question [99] for theory validation is: "Does the model make a valid use of the theory?" This validation type is arguably the most difficult to present a clear case for. The previous validation types of agent-based models are much clearer in what is being described and claimed. One can easily observe a flow chart and understand the totality of what is occurring. This can then be compared with natural phenomena and the parts that are not being modeled or abstracted away can be discussed. This ends with either agreement or disagreement on the validity of a concept or behavior. There may be disagreement, but at least the disagreement can be pointed to. The same could be said about input, data, and output validation. Theory validation is much more abstract. The main theoretical research area of Cyber-FIT is computational and mathematical organization theory. A definition of this theory is: "Computational and mathematical organization theory is an interdisciplinary scientific area whose research members focus on developing and testing organizational theory using formal models. The community shares a theoretical view of organizations as collections of processes and intelligent adaptive agents that are task oriented, socially situated, technologically bound, and continuously changing " [111]. This is exactly what Cyber-FIT sets out to do, most specifically modeling the cyber team as adaptive agents. Each part of that theory definition can provide clarity on how Cyber-FIT does make a valid use of computational and mathematical organization theory. This is the over arching theory that this model aims to extend and contribute to. When moving down a level to its many sub-components, other theories can be integrated and validated as well. The table below provides a description of each of the theories that are validated within Cyber-FIT.

Overarching	
Theory	*Agent Classes*
Computational and Mathematical Organization Theory	Defender, Terrain, Attacker, Friendly, Interactions
Sub-component Theories	
Theory	*Agent Classes*
Cybercrime	Attacker
Cyber Situation Awareness	Defender
Performance Theory	Defender
Network Science	Defender, Terrain, Interactions

Table 51: Theory validations within Cyber-FIT model

Speaking from a cyber team perspective, the team is made up of intelligent adaptive agents. They keep track of what is occurring on the cyber terrain they are interacting with, providing a simulation of intelligence. The cyber team agents are task oriented. They are always working on a specific defensive cyber operation with a goal. Once the goal is reached, they move on to a new task. They are socially situated, that is they will communicate with other team members to pass informational messages. They are technologically bound because of the agent rulesets that determine what they will or will not do. Finally, the cyber team agents are continuously changing. On every tick they either continue their current operation (updating their own cognitive model of the terrain agents), getting a new operation (based on what is occurring in the environment), working to restore compromised terrain agents, or doing nothing (simulating and tracking stuck time). All in all, this model, holistically, is primarily planted in computational and mathematical organization theory.

5.6.1 Sub-component theories

Moving on to the sub-component theories, as shown in the table above, three have been integrated into Cyber-FIT. The first is cybercrime, which was the driver of Cyber-FIT version two. In version two, the attacker agents were upgraded to force their behavior through the cyber kill chain. This is a simulation model of an aspect of cybercrime. Gordon and Ford define cybercrime as "any crime that is facilitated or committed using a computer, network, or hardware device" [112]. They go on to differentiate two types of cybercrime where type I is associated with technology and type II is associated with the human element. Cyber-FIT version two could be described as modeling this type II aspect of cybercrime. The cyber kill chain itself, is an attempt at merging the technological and human aspects of what nearly always must occur to compromise a computer system. In order to better understand the human aspects of cybercrime, many have contributed to adversarial modeling, tangentially related to this work. For example, states of adversary

behavior can be simulated, and or observed which creates graphs. This can be used to build simulation models for security system evaluation [113]. Cyber-FIT can be extended to simulate increasingly complex attacker behaviors and then outputting many different data temporally.

Cyber situation awareness theory was covered in detail in chapter 2.3. The working definition used by Cyber-FIT of cyber situation awareness theory is given by Onwubiko [50] as "processes and technology required to gain awareness of historic, current, and impending (future) situations in cyber". This drives the agent and team level computations of cyber situation awareness. As described in chapter 2.3, cyber situation awareness is measured as a function based on knowledge (past), comprehension (current) and projection (future), of the cyber operations. As of this writing, situation awareness is not a measurement that cyber teams discuss quantitatively, it is an abstract concept. This work provides a simple mechanism for simulating cyber conflict, and specifically defining the data that could compute cyber situation awareness. Situation awareness has always been difficult to measure, even in situations where relevant knowledge is much clearer. A famous example is the work of Endsley to query fighter pilots and determine at different points in a simulated mission if they could pinpoint enemy locations [114]. Cyber-FIT can be used in a similar fashion with military cyber teams. This work contributes to the field of cyber situation awareness theory by 1) creating a novel metric computationally defining cyber situation awareness and 2) providing a software framework to experiment with that definition or extend it.

Performance theory is also incorporated into this model as the current version output measures are defining the performance of the simulated cyber team. Measuring performance is typically situational [115] and teams with a specific common goal are usually easiest to measure. Work has been done using surveys for teams with tasks more difficult to define [116]. Cyber team performance measurement has been studied recently by various works. One study used a combination of self-assessments, exercise data, and observer data to compare teams . This study identified several performance measures that are very similar to those simulated in Cyber-FIT including "attack discovery", "vulnerability removal", and "DMZ attack success rate" [117]. Cyber-FIT as constructed can support all of the measurement types and styles proposed in each of these works. Due to the object-oriented nature of the software, each agent can be instrumented, so to speak, with any imaginable data structure. For example, if the performance metric depends on a new data definition, it can be added to the team object, individual agent object, or some number of the terrain objects.

The last sub-component theory is network science. A key design decision for the architecture of the framework in version one was to link the agents. This proved very useful, especially in version four where the object-oriented nature of the model made it easy to collect link information temporally. As shown in chapter four, two network science

measures were successfully computed for the entirety of the simulated conflict: Force-Force Interaction Network Node Total Degree Centrality and Terrain-Terrain Interaction Network Density. Again, a primary contribution of this work, is that based on the current version, any network science measure could be analyzed as it relates to the links connecting agents. This could be communication networks, computer architecture networks, attack graphs, etc. Cyber-FIT is now well suited to carry out simulations of organizational change. An illustrative example is a simulation where the trend in stability changed as number of employees was increased for an organization [118]. Cyber-FIT could be used to determine if this behavior can generalize to cyber operations communications and learning within the organization.

5.7 Validation conclusion

In summary, this model has been validated, in some way, using all of the validation strategies described by North and Macal [99]. The art, rather than science, of model development is the key driver for validation decisions. Cyber-FIT began as an abstract concept drawn on a whiteboard. Once version one was completed, it was clear that there was something interesting and novel in place. Version two was the first validation attempt as the model was tuned to simulate ranges of outcome variables that matched the empirical data provided by the Alphaville exercise [46]. Version three incorporated the first specific theoretical validation by incorporating cyber situational awareness. Version four was a large overhaul and re-architecture of the software and included all validation types described in this chapter. It is difficult to clearly articulate precisely what parts of a model are validated, in what way, and by how much. This is a difficulty encountered by nearly all software developers – the conceptual model differs from person to person. This is precisely why the validation in parts was described systematically, type by type, in this chapter. A summary of this can be shown, visually, in different ways, depending on how the parts of the model are categorized. The next two figures are representations of the totality of validation in parts for the entire model.

Behavior Data	Requirements Validation	Data Validation	Face Validation	Process Validation	Output Validation	Agent Validation	Theory Validation
Terrain Vuln. Growth Rate	☑	☑	☑			☑	
Defender K, S, E	☑	☑	☑			☑	☑
Defender Cyber Ops	☑	☑	☑	☑		☑	☑
Defender Interaction Rate	☑	☑	☑	☑		☑	
Attacker Phase Time	☑	☑	☑	☑	☑	☑	☑
Attacker Zero-Day Rate	☑	☑					
Attacker Tier Level	☑	☑	☑				
Terrain Configuration	☑	☑	☑				
Mission Cyber Ops	☑	☑	☑	☑		☑	

Figure 62: Behavior data validation in parts

Output Data	Requirements Validation	Data Validation	Face Validation	Process Validation	Output Validation	Agent Validation	Theory Validation
Terrain Vulnerability Rate	☑		☑	☑		☑	
Terrain Compromise Rate	☑		☑	☑		☑	
Time to Detect	☑		☑	☑		☑	
Time to Restore	☑		☑	☑		☑	
Operational Efficiency	☑		☑				
Cyber Situation Awareness	☑		☑				☑
Cyber Mission Capability Rate	☑		☑	☑			
Time to Compromise	☑		☑	☑		☑	
Compromise Rate	☑		☑	☑		☑	
Network Measures	☑		☑		☑		
Mission Measures	☑		☑				

Figure 63: Output data validation in parts

6 Chapter 6: Conclusion

The Cyber-FIT simulation framework version four represents an evolution of capability leading to a more realistic representation of cyber team performance. In its current form, realistically scaled simulation of cyber conflict can be conducted, and the object-oriented nature makes for easy extensibility. Returning to the original goals of this thesis, they are: 1) define cyber team performance measures, 2) create an agent-based software framework to simulate performance outcomes, and 3) validate the software. All three goals have been achieved.

Cyber team performance measures, even at this writing, remain a controversial topic. In fact, this past Cyber Flag exercise included the first ever crowning of a "champion" [119]. According to the article the team did best at analyzing "intelligence regarding threats and malicious actors, conduct mission planning, 'deploy' to the compromised network, detect malicious activity and recommend response actions to the mission owner during an evolving event scenario". This thesis was started in 2017, five years ago, when I personally attended Cyber Flag and talked with cyber protection team members about how hard it would be to crown a champion of Cyber Flag. Now, five years later, the first champion was crowned, with a scoring system informed in part by my work. This was a wonderful development and very enjoyable way for thesis work to be utilized in the real world.

Also, my work at the Software Engineering Institute continues to focus a large part on how to assess, evaluate, and compare cyber teams. To put it simply, there is not a consensus amongst the field! A sports analogy will do well to illuminate. When a professional American football team plays a game against another team, they play on a standardized field, with consistent rules, and a well-established scoring system. When the same team practices, their practice field (typically close by to the game-day stadium) is exactly the same as the game-day field. The field size, the turn, the markings are all the same. When they have a practice game, they bring in referees to score the game just like real games and play by the same rules with the same scoring system. They'll even spend resources simulating different weather possibilities and pumping in simulated crowd noise. They'll replay parts of the game that are most difficult and practice those over and over in the same way that they'd expect to encounter in a real game.

This is exactly what cyber teams want. They want to hold practice games in a simulation of the same exact environment (field), with the same constraints (rules), and with the same stakes (scoring) as their real-world operations. This does not exist. This work helps move towards that ideal state by providing a realistic, vetted, computational, and validated set of cyber team performance measures that should be used in the simulator that cyber teams are requesting.

The second goal of this work was to build an agent-based software framework to simulate the performance measures of cyber teams. This work started in 2016 when the

first lines of code in NetLogo were written for the first version. The first working prototype simply had agents searching for vulnerable and compromised states of terrain agents, all the while attacking agents were trying to compromise vulnerable agents. This simplistic first version was enough to show subject matter experts and mostly agree that this was the lowest level representation of what was happening in the real world: Terrain agents must be vulnerable. An attacking agent has to find and exploit that vulnerability. Defending agents have to find and fix that vulnerability. That's the lowest level. This agent-based model evolved to add the complexity described in Chapter 2.1 marking version one of Cyber-FIT. Once Cyber-FIT was created, it wasn't apparent that the focus of the model would be cyber team performance measures. At first, I was simply trying to break down the nature of cyberwar to its lowest level. Once version one was complete, the idea switched to determining what exactly would this model provide, from a scientific software perspective.

Also, being that this is a Societal Computing doctoral thesis, I wanted to address an issue that was widely recognized. Using Cyber-FIT to simulate network performance of various cyber terrain architectures, while interesting to a subset of cyber professionals, would not map to a larger societal issue well. This is why my experience at Cyber Flag 2017 was so informative. Cyber Flag 2017 was an excellent and impressive exercise. But unlike Cyber Flag 2021, there was no champion crowned. Being that I was working on another project at the time where we were trying to create a "situation awareness" quiz for cyber operations, I was picking the brains of cyber operators at Cyber Flag 2017 about assessment and performance in general. This led to conversations about what the tactics of cyber operations are and how each team member contributes. Furthermore, it was hard to find manuals or operating instructions that would describe how to conduct an operation, and then what the performance measure would be, thereby understanding how well a team did. In totality, this thesis works through difficult questions like these one by one, learning things along the way. This can be summarized with the following table which summarizes all of the virtual experiments conducted during this work. An indication of the academic contribution this model makes is the differences in the insights that can be achieved from a framework type model.

VE	Research Question	Insights
1	How many forces should we deploy to minimize the effect of a routing protocol attack (RPA) in an industrial environment?	Simulate the assumptions of your security posture to quantify differences
2	What will be the expected effect on cyber terrain if the adversary switches from a fifteen-day routing protocol attack to a denial-	Incorporate indicators of compromise into planning

	of-service attack in a base environment with six troops deployed	
3	What number of forces maximizes expected cyber terrain mission capability rate against random attacks in a tactical environment?	Baseline organization security operations against a distribution of potential attacks
4	Can the model simulate attackers moving through the cyber-kill chain in accordance with the empirical data?	Model tuning can lead to realistic simulations
5	How many DCO forces should be deployed to maximize the time to complete phases three and four during a routing protocol attack with an exploitation success rate of 15 percent?	Expectations of adversary behaviors can be quantified
6	How many user systems will be compromised as phishing attack targets are increased?	User training security policy can be simulated
7	What is the average time to complete a routing protocol attack with eight DCO forces deployed, as the vulnerability growth rate increases?	Minor changes to cyber security models have big effects on outcomes
8	What is the expected time to complete phases three and four, during a denial-of-service attack, with six DCO forces deployed, as the exploitation success rate is increased?	Emergent behavior can often be the opposite of intuition
9	What is the maximum cyber situation awareness during a cyber team survey?	Cyber missions are ideal candidate for agent-based models because of defined desired outcomes
10	How long does it take for a cyber team to obtain maximum cyber situation awareness?	Quantifying a theoretical cognitive model can help shape mission goals
11	How much better will a highly skilled cyber team perform then a medium skilled team, against all six adversary tiers?	Cyber training and retention efforts should be simulated for cost benefit analysis
12	If a cyber team is delayed to the conflict, how quickly can it recover compromised systems against all six adversary tiers?	Wargaming simulations of cyber terrain can be fed by output data from models like Cyber-FIT

Table 52: Summary of virtual experiments

Document	Wording	VE
2013 Dept of Defense Science Board Report	"The Task Force could not find a set of metrics employed by DoD or industry that would help DoD shape its investment decisions. A qualitative comparison of resources and DoD level of effort in relation to the success rate of red teams is clear evidence of the ***lack of useful metrics*** "	1, 2, 7
2015 Dept of Defense Cyber Strategy	"develop the data schema, databases, algorithms, and modeling and simulation capabilities necessary to ***assess the effectiveness of cyber operations***."	3, 5, 6
2019 White House Executive Order	"goal of the competition shall be to ***identify***, challenge, and reward the United States Government's ***best cybersecurity practitioners and teams*** across offensive and defensive cybersecurity disciplines."	4, 8, 9
2020 Air Force Broad Agency Announcement	"having very specific and valid ***knowledge, skill, experience, and performance*** (K, S, E, P) information on what airmen in various roles are expected to be able to know and do and how they are ***expected to operate and perform***"	10, 11, 12

Table 53: Mapping of requirements to virtual experiments

This basic difficulty still exists. There are no industry recognized guides to cyber team performance showing quantitative benchmarks. Returning to the American football analogy, teams know precisely how fast each team member runs, and how far a passing play should cover. This is the standard we should expect for our cyber team leadership and policy in general. We have to know what is occurring in cyberspace, at a precision that will allow us to continually get better. This is a worldwide security imperative as more and more of our society is conducted in cyberspace.

The third goal of this thesis was to validate the software. This was done in all of the ways generally accepted amongst agent-based practitioners. The North and Macal methodology was followed, type-by-type to conduct an overall validation in parts. A key contribution of this work was to conduct a survey and focus group. The survey validated data used for simulations in general, and the mismatch of understanding amongst professionals in the field with how they understand their own performance versus their team performance. The focus group validated the software requirements, the agent processes, and the ultimate goals of this work. Skill is the x-factor that government and

industry leaders should focus un, this is unanimous. Now, the next step is to figure out how to improve the collective skill of the cyber forces that our security depends on.

6.1 Implications for Human Cyber Team Training

When this model was envisioned and then developed, tested, tuned, and used for simulations, it was done so in a general sense where are all of the measures might be useful for any generalized cyber team. When discussing the current model with cyber security experts, it turns out that in many cases training will be specific, and not generalizable. Typically, teams will focus training on the activities they are most likely to encounter in the near term. This means that an army regional cyber center, overseeing many different locations and processing data feeds from those disparate units will have different operational and training goals than a team that works for a regional banking security operations center.

Another consideration realized upon conclusion of this work is that organizational cyber security improvement is usually executed differently through individual training and team training. Individual training is simpler to execute and cheaper. This is because individual training can be accomplished with far less disruption to the team's normal operations. A team of ten can continue operations just fine while one of the members takes a three day course, for example. That same team of ten can not continue normal operations if six members complete a three day team-based course. Also, team-based training is typically more expensive and much more difficult to find. A simple internet search will show dozens upon dozens of private companies offering individual courses to improve a wide variety of cyber security skills. This is not true for team-based skill improvement.

This means that this work informs organizations by showing what can be reasonably quantified from a skill improvement perspective and how improved performance manifests. So, if an individual cyber security training course improves overall knowledge of how networking system vulnerabilities can be managed and patched, then how would that performance improvement manifest? Based on this work, it would mean that the organization would see an improvement to the complexity of tasks that the security team can successfully manage thereby lowering quantified vulnerability level. The organization can consider team level improvements in a similar fashion. If the organization sends a small team to an incident response communications exercise, the communication network should show improvements to its structure over time. The time to close an incident (using ticketing systems) would be lower moving forward because of the practice with whom talks to whom, and where responsibility lies.

Ultimately, this work tells organizations that they should carefully calculate what it is the improvement they are searching for improves organizationally. This is always situational and team specific. The first step is to write out specifically what measurable is should be improved and how that measurable is currently calculated. This step alone is

enlightening because the organization, many times, will realize it isn't considering a measurable because it's not measured. Once specific measurables are identified, the next step is to determine which knowledge, skills, and/or experience will improve that organizational measurable. This is mapping the inputs (your team's abilities) to the outputs (your team's measurable effects). Again, the mapping in and of itself is informative and constructive for an organization to work through. The last step is to collect data that provides insight to how the inputs and outputs have changed. This step is by far the most difficult due to a number of issues. The data might be operationally sensitive, so it would have to be safeguarded (resource intensive). The data might be individually sensitive which means that employees might feel invaded by looking at specific details about their efforts. The organization could seem overbearing or micro-managing, so that must be navigated. The data might be incorrect. For instance, when collecting output data about system outages (attempting to minimize downtime) a network error could be confusing the outage, or the dashboard could be misconfigured. All in all, there is a cost to measuring, so this cost should be minimized to maximally increase performance. Unfortunately, it seems that as of this writing most organizations informally err on the side of very little to no measuring, which leads us back to the insights from the focus group. There is little to no quantification of cyber team performance throughout the industry.

6.1.1 Metrics to add to human team training

It is informative to consider what was done within a simulation model and then envision what can actually be accomplished in the real world. Part of the art of this work was the line to draw when coming up with performance measures. Obviously, there could be many more measures added, simulated, calculated and discussed. The final list was what was most often discussed amongst practitioners in the field, covering all of the most pressing needs. Some measures might be considered difficult to map to outcomes such as transactive memory. Others would be harder to measure like cognitive adaption over time. But there are clearly some that could provide insights about how cyber teams perform. If an organization wanted to start a cyber team training program, this work implies that the following metrics are most suitable, realistic, and impactful to be added at this time. Also, the only way that cyber team training can be done, is within a virtual cyber range. Cyber ranges are widely available in many forms, so all of the metrics mentioned here is assuming that the range provider can instrument the range in a way where data can be extracted that is relevant to the particular training goal. This is a trivial matter as of this writing as ranges are completely configurable to pull nearly any data imaginable out of them. The following measures should be considered to be added for training, first individual and then team-based.

6.1.1.1 Individual metrics

Fatigue – This is the error rate as a function of time. As the training engagement goes on, the range can report on errors being made by the individual trainee. This is an indication of fatigue for the individual. This error rate can be used for simulation models which can inform virtual experiments concerned with fatigue affecting mission outcomes.

Stress – Stress is an important individual measure that will affect how the individual performs tasks in terms of selection, speed, and accuracy among others. Stress can be introduced using injects that can cause concern for the trainee. This could be in the form of requiring a report on an unknown system that must be done in a time frame too short for the trainee to be reasonably expected to complete. Cyber ranges can be instrumented with technology such as key logging that would show a burstiness indicative of stress.

Communication – Communication scores can be thought of in many different ways at an individual level. One way would be to grade the quality of reports sent back to the simulated commander in a training lab. Another way is to grade the communication based on speed. That is – once a cyber attack is launched how long until the trainee creates report in the incident ticketing system. Cyber ranges can easily extract this information in a format that can be automatically graded.

Cyber Trustworthiness/Confidence – Measures that determine seemingly qualitative characteristics can also be tracked and added to individual cyber training systems. These are characteristics that are discussed amongst practitioners, but have no formal definition within training manuals or official government/military sources. Yet, they exist. A cyber range with individual training labs can glean information to understand metrics such as trustworthiness – is the trainee answering quiz questions that they don't actually know the answer to. Or, cyber confidence, which can be thought of as how sure the trainee is of the steps they are taking within a tasking.

6.1.1.2 Team metrics

Terrain vulnerability/compromise rate – These metrics are clear to all participants in what they are trying to accomplish. This metric should be minimized. Also, there is a distinct tie to what actions they take and how this metric changes. Within cyber training exercises, the team typically sets out searching (hunting) for vulnerabilities or indicators of compromise. The steps they take to accomplish these ends are easily observable, especially if incident response systems show detailed steps taken and records of evidence collected. These metrics are easily calculated within the context of cyber team training exercises. The control group knows what vulnerabilities are present in the virtual range and which systems are compromised. This data, changing over time through the entirety of the exercise shows the team their measurables at any time.

Time to detect/restore – These metrics are also very clear within a cyber team exercise. The team is working together and coordinating activities to minimize the time it takes to detect malicious activity and then restore or mitigate that problem. In cyber team

exercises the control group must track these data to get the ground truth of when an incident or attack was realized. Then, the control group must have a way so that the timestamp can be collected when the team detects and restores the compromised systems. Again, incident response ticketing systems are an ideal candidate because the team engaged in the training is incentivized to self-report those actions and evidence as quickly as possible to receive favorable scoring.

Cyber situation awareness – Although this measure is considered more theoretical at this time, that is due to the potential disagreement with precisely how to define it. This model does defines it is a function of knowledge, comprehension, and projection. An individual organization can easily determine what their own, most pressing definition of cyber situation awareness is. What is most important to know, comprehend, and project? Then, within a cyber team exercise, there are ways to determine if the team is on the right track. The easiest way is to give pop quizzes that ask for information important to the current challenge. This is an easy way to measure both knowledge and comprehension. Probing questions about what should be done next could be asked and scored by the control group. The team will be seeking to score as many points as possible when the situation awareness quizzes are presented.

Cyber mission capability rate – This measure would be calculated in a cyber range by defining key terrain that must be protected and then tracking requests and fulfillments to/from those systems. This can be done with packet capture software or Netflow sensors. Determining the speed at which the information is transmitted might be difficult. To get around this the rate can be baselined the day previous to the exercise, and then the team would be expected to keep the rate at the baseline operation. When the control group instructs the red team to begin attacking systems, the team should seek to maximize the cyber mission capability rate.

Communication efficiency – This measure would be calculated by inspecting electronic messaging systems within a cyber training exercise. Most exercises have instant messaging capabilities embedded in the range so that trainees can communicate easily while defending cyber systems. Communication efficiency would be team specific, but in general the team would seek to maximize this metric. Training providers would have to score each message on content where messages that are related to information leading to positive operational activities would be scored higher. This can be compared with messages that are either detrimental to positive operations or unrelated.

Communication network density – This measure would also depend on extracting data from an electronic messaging system such as instant messaging within a cyber range. This metric would be easier to calculate then communication efficiency because there would be no need to make score the messages which is resource intensive. Instead, the control group simply has to determine what time periods should be analyzed for a dynamic network analysis. This could be the interaction network over days, hours, half-days, etc.

Similar research has been done tying network density measures to successful hospital team performance [120].

Key entities – Another measure that would be calculated using extracted communication network data is key entities. These are dynamic network measures that one would use ORA to run analysis leading to agents that are most important for information flow. This is commonly referred to as "informal leaders" – those that are making a big impact even though the organizational chart doesn't specify an official leadership position. Generally speaking these measures should be the higher the better when looking at node metrics. Simply extracting the interaction network data from electronic and instant messaging systems from the range is that would be needed for range instrumentation.

Tool Scoring – Investment decisions are made all of the time in regards to the question: what tool should we buy? This means that, essentially, management thinks the cost/benefit analysis points to one tool over another. This is a perfect virtual experiment research question and can be seen as a team measure. The tool can be scored in terms of relative performance comparisons. This same concept can be applied to network topologies and architectures.

6.2 Limitations

Cyber-FIT is attempting to model an extremely complex real world system. Most complex system simulations will begin like Cyber-FIT did: the lowest level, most basic behaviors first, and then add complexity along the way. The limitations of Cyber-FIT will be presented first as compared to one of the original design considerations, modeling a cyber conflict in order to understand what it is precisely, and computationally, a cyber team is trying to affect. Second the limitations in terms of human behavioral modeling will be discussed in detail due to the importance that this aspect represents. Third the limitations with how attackers behave and affect vulnerabilities is discussed. The current state of Cyber-FIT is working software that can be extended in many different ways. Laying out the limitations in this way naturally leads to the next opportunities for development of new versions.

6.2.1 Cyber conflict modeling in general

Returning back to the introduction of this thesis, recall Table X that laid out all of the input, behavioral, and output data categories that would be necessary to realistically simulate a cyber conflict so that cyber team performance could be quantified. This listing represents a best-case scenario of data and capability. In concluding this work, an assessment of what Cyber-FIT provides is informative. Taking stock of what Cyber-FIT was not able to address is in effect a display of the limitations of this work. Figure 64 below assesses the current progress of Cyber-FIT, and what level of modeling is achieved

per each simulation requirement category. For each simulation requirement, the columns are conceptual, computational, and validated. Conceptual means that the basic architecture is available to extend the software to encompass that category. Computational means that the software is currently differentiating that input or outputting data that can computationally define the behaviors of interest. Validated means that the input, behaviors and/or output have been validated as defined in chapter five.

Model Behavior/Data	Conceptual	Computational	Validated
Base Cyber Terrain	☑		
Kinetic Missions Supported	☑	☑	☑
Cyber Team Rosters	☑	☑	☑
Adversary Intelligence	☑		
Cyber Policy	☑		
Defender Cyber Operational Behavior	☑	☑	☑
Attacker Cyber Operational Behavior	☑	☑	☑
Friendly Force Cyber Operational Behavior	☑	☑	
Human Behavioral Modeling			
Cognitive Modeling	☑	☑	
Cyber Terrain Network Behavior	☑		
Performance Measures	☑	☑	☑
Network Data	☑	☑	☑
Modeling Environment	☑	☑	
Data Collection and Processing	☑	☑	☑

Figure 64: Ideal cyber conflict simulation software categories of input/behavior/output

Walking through each of these software simulation categories and columns serves as an analysis of Cyber-FIT limitations. To begin, category number one is base cyber terrain. There is always a data center with the most critical computing, network and serving needs for an organization. Edge routers connect to the internet and with computers inside the corporate network. Servers host webpages and databases that provide systems for use by employees and services such as identity management. There are security devices and software throughout this core infrastructure. Cyber-FIT conceptually models this activity by enforcing a base infrastructure to exist independent of the systems that the kinetic mission forces utilize. The base architecture is three-tier in nature, each system being one of three types (networking, serving, hosting). This behavior is not computational within the model at this time. The differences in infrastructure does not make a difference in how the any of the cyber team performance measures are computed. The infrastructure terrain agents do behave in the same way as the mission terrain agents, and the defending agents will survey, secure, and restore compromises on those terrain agents as well.

The second simulation requirement is the details and behaviors of the supported kinetic missions. The primary purpose of cyberspace is to move data through systems that ultimately provide information to kinetic mission operators. For example, the soldier in the field using a voice communications device to get updates from the command post. Cyber-FIT does load a kinetic mission file that provides number of personnel and number of cyber terrain by type. The personnel are the friendly agents that make information requests to the cyber terrain they are associated with. This provides an excellent baseline behavioral mechanisms with which to apply computational measures such as cyber mission capability rate and the network measures differentiated by mission. These behaviors and novel measure have been face and agent validated.

The next simulation requirement is cyber team rosters. This is the demographic details of cyber teams in terms of experience, knowledge, skill, certifications, previous missions supported, specialty knowledge, leadership training, etc. This aspect is conceptually modeled into Cyber-FIT within the defender agent class. Each defender agent can have any number of attributes added on with member variables. These member variables then affect behaviors, leading to differential performance. Therefore, this category has been conceptually and computationally achieved. It has also been through several of the validation types through the survey and focus group.

Next is adversary intelligence information that would be of interest to the cyber team. In real world operations details like this are in the form of expected adversary details, affiliations, recent activity, indicators of compromise, and reporting instructions for contact. As of this version Cyber-FIT only conceptually models adversary behavior by assigning a tier level to the attacker agents and allowing for any number of adversaries to be added to the simulation. Complex agent rulesets could easily be added such as how to handle expected adversary indicators of compromise (defender agent already has permission to remove) versus unexpected (defender agent has to request permission to clear).

Cyber policy data was a candidate for modeling but was not included in the current version of the model. Details of cyber policies could be added to the terrain agent class in forms of security hardening and versioning. Another way to add cyber policy data would be through defender agent class rules dictating how they are to interact with both other defender agents and terrain agents in completing cyber operations. Version one of the software did use environment type to simulate the differences in where vulnerabilities occur at different rates. So, conceptually, cyber policy details have been conceptually accomplished and the framework can support moving on to more complex investigation.

Defender cyber operational behavior was the most thoroughly studied and modeled aspect of Cyber-FIT. The defender agents have the most complex behaviors in terms of how they select operations, carry out operations, communicate with each other and are instrumented for data collection. The survey, focus group, and countless conversations

with subject matter experts was primarily focused on how to find the right level trading off simplicity and complexity for a robust framework included in this version of the software. This category has been thoroughly conceptually and computationally modeled. Many of the validation types in chapter five were completed on the defender agents.

Attacker cyber operational behavior was also fairly thoroughly studied and modeled within Cyber-IFT. The attacker agents must traverse the cyber kill chain according to realistic time scales from empirical studies. Like in real life, each phase must be successful to continue on toward their goal of compromising systems and taking actions. The attacker agents are conceptual, computational, and validated. The attacker agent class is instrumented to collect data in various ways that allow for virtual experimentation. The main behavior that was not included in this version was different attack types by the attacker agents. For instance, how to model an exfiltration attack differently than an advanced persistent threat. This behavior will be discussed in the future research section.

Friendly force cyber operational behavior was conceptually and computationally modeled in this version of Cyber-FIT. One of the key cyber team performance measures defined in this thesis is cyber mission capability rate. This measure depends on friendly force agents utilizing cyber terrain. This requirement drove the conceptual and computational modeling. The friendly force agents are assigned teams (by mission) and data is collected on their information requests and how quickly the request was fulfilled (if at all). This data determines the defender agent's cyber mission capability rate. This behavior has not been validated. A candidate for this validation would be using log data from web requests to see how often these requests fail.

Human behavioral modeling was not included in this version of Cyber-FIT and this is one of the biggest limitations of the software. All three classes of humans (defender, attacker, friendly) operate the same every tick of the simulation. The agents do not forget, or sleep, or change shifts, etc. Mistakes are modeled into the defender agent class in the form of agent rulesets based on knowledge, skill, and experience. This is the only time the human agents do not operate perfectly. The friendly agents don't operate differently due to the nature of their mission. The attacker agents do not change tactics based on changes to the operational environment. There is no transactive memory amongst teams. All of these are examples of limitations and places where the software could be improved.

Cognitive modeling is conceptually and computationally modeled in this version. The concept of cyber situation awareness was identified as an early development goal within Cyber-FIT. This is because cyber situation awareness is a well known yet not well understood problem in cyber operations research. Situation awareness is cognitive, so conceptually, this is modeled as soon as the software tracks anything that the agents are "thinking". Cyber-FIT measures situation awareness per agent and per team in a novel way. The measurement includes actual knowledge of what is true in the environment, based on the agent's cognitive model of the security status of the terrain agents. It also

includes a more abstract values about what is impending based on how well the agent is anticipating its next move. This measure has not been validated because of how difficult that work would be, it is out of scope for this study.

Cyber terrain network architecture is only conceptually modeled in this version and is a limitation. The base cyber terrain is deployed as a three-tier network architecture in terms of realistic numbers of networking, serving and host machines. However, none of the machines communicate with each other according to realistic network architecture rules or routing protocols. Similarly, attacker agents do not have to traverse a network route to search for vulnerabilities or attempt to compromise a terrain agent. There was never a point in defining performance measures where this complexity was needed, so it never became a requirement. This type of complexity would be a good candidate for near term improvements.

Performance measures, being the primary design goal of Cyber-FIT, are conceptual, computational, and validated. One of the primary contributions of this work was the subject matter expert testimonials, survey, and focus group which holistically validated the performance measures. The only limitation of this part of the work is in the data processing aspect. The software tracks all of the data necessary to compute all of the measures for every run of the simulation. But the user must specify the data to track in the Repast interface and then post-process the data oneself. An agent-based model itself, generally speaking, and Repast specifically does not typically provide an output interface. This is because the output is contextual. The user must determine how to interpret data such as "vulnerability level" and apply it to the scenario in question. Also, in a temporal simulation, time must be defined. In this thesis, each tick was always simulated as a minute, but that need not be the case for all experiments. Finally, network data that connects nodes and links are output and have been validated with this version.

The modeling environment refers to the framework in general. For all of this to work together, there must be a software defined system that ties all of the previous mentioned requirements together, or else it's a federation of software, which would not be useful. Cyber-FIT and its definitions based on Repast libraries is the modeling environment. In a sense the User Observer Class is the glue holding all of the other classes together and instantiating the interface definitions that enable connections amongst the objects in memory at compute time. This modeling environment is conceptual and easily extendible so that new concepts can be added in. It is computational in through automatic data collection. There is no way to validate the modeling environment itself. The most pressing current limitation of the modeling environment is the need to define inputs through files. An early development goal was to use the interface itself to set parameters of interest and quickly re-run simulations with changes that can be seen with the output visualization. This was difficult to code and configuration files were used early on. This feature has not been improved as of this version.

Data collection and processing has coded into the software in the form of class methods and variables. Each agent has a number of methods that either define behaviors, or update member variables to track data at every tick of the simulation. Once the user interface is initialized, countless combinations of data can be called using the built-in Repast processing features. In this way the data collection and processing requirement has been achieved in a conceptual, computational, and validated way. The limitation of this aspect of the software is the effort required to conduct data analysis. Repast data collection functionality is not easy to use or troubleshoot. The data will be provided in separated value format (comma, tab, space). It's up to the user to then clean and define the performance measures defined in this thesis. At the end of a run of the simulation, there is no report that defines, for example, the cyber mission capability rate. Instead, the user must do that. For all of the simulations I have run, and virtual experiments I've conducted I use some combination of Python, R, and Excel.

6.2.2 Human behavioral modeling

The primary concern with the current version of Cyber-FIT is the lack of human behavior within the defender agents. At the beginning of this work, it was a goal of mine to make the agents human-like in some way. As the work progressed and goals changed human behavioral modeling took a back seat. This was mainly due to the amount of effort and focus taken on creating the dashboards, analysis and virtual experimentation presented in chapter four. Throughout that work there was never a point where a specific need for more complex human behaviors be incorporated. It might have changed what the trends or results of the output measures were, but it wouldn't have changed the output measures themselves. Now, at the end of version four, several areas of human behavioral modeling can clearly be incorporated into either the next version of Cyber-FIT or versions soon enough. These behaviors are grouped into skills, cognitive, and organizational, and will be discussed next.

6.2.2.1 Skill is the X-factor

Skill being the X-factor was a very interesting concept that I seemed to stumble across within this work. It is an element of cyber teaming that came up over and over again. I remember well a cyber war exercise I was participating in where the team was not doing well. We in the white cell were observing and discussing their poor performance. The team had the typical amount of personnel and the typical amount of requisite knowledge, skills, and experience, on paper at least. They just weren't "getting it". On day three, the officer in charge brought in a new non-commissioned officer in charge (NCOIC) to act as cyber team battle captain. Within two hours the team was performing at a much higher level. They were communicating specific vulnerabilities and indicators of compromise that were present on the range and relevant to the simulated adversaries.

By the end of the next day the team was able to find all of the relevant advanced persistent threats, upgrade the security posture of the network and deliver an excellent out brief to the commander. It was clear that the entire team was better off, deriving a lot of training value.

What changed? The new NCOIC, of course. This leader was able to better direct the troops and coordinate complex tasks. He saw where the holes were and who was not sure and was able to push them in the right direction. This is the X-factor at work. Modeling, defining, computationally simulating the X-factor is extremely difficult but both interesting and necessary to understand the true nature of how skills affect cyber teams. This also shows us that "skill" is not one size fits all label or characteristic. Skill can be broken up into many different sub-components. This NCOIC was displaying "leadership skill" or in more traditional military nomenclature, "troop-leading skill". Defining different types of skills might actually be helpful for computational modeling. Consider the leadership skill category. This would mean that in a computational model, the software agents would be more efficient in sending information or discovering agents that are stuck. Another example could be "tool skill". Cyber teams all have toolsets which is the software and hardware used to do their job. Tool skill would be level of mastery with using those tools in a simulated environment where mastery would manifest quicker, and mistakes with sensing evidence would show less.

This expanded notion of skill would help with mapping simulation to real world data. In the cyber range, during a team training exercise, the white cell could provide different tools and then observe the differences in how different teams use them to hunt for an advanced persistent threat. This is an easy simulation to develop and run. The agents, using the tool, would also hunt, moving through the terrain agents at different speeds, error rates, access rules, etc. Comparing simulation data side-by-side with exercise data extracted from the range would illuminate tool skill definitions and what is quantifiable, actionable, and achievable.

6.2.2.2 Cognitive modeling

The cognitive model of all three agent types is an excellent candidate for improvement. The first to improve would certainly be the defender agents. As of the current version they act too much like robots, following rules and instructions. There is some stochastic simulation occurring in terms of what operations they take on, how long they take to complete those operations, what operations they decide to act upon, and when they communicate. But, all of these agent actions could be informed by cognitive limitations. So, how could this be approached? We can draw from work laying out a "standard model of the mind" which is broken into four parts: structure and processing, memory and content, learning, and perception and motor systems [121]. This model can be emulated within the defender agent class to simulate those four parts of the cognitive architecture while working through cyber operations.

One way this will work is by updating the data structures representing memory of terrain status so that those worked on most recently will be more likely to be recalled correctly and those that haven't been worked with will begin to be removed from memory, simulating forgetting. When an agent is attempting to work with a tool to secure terrain, this is an example of processing and perception interacting to then result in correct or incorrect steps being performed.

A major issue with cyber teams is fatigue that changes over time (agents getting tired) and increases under stress (incidents and reporting requirements. When an agent is under stress (which can be a Boolean or distribution of values), it should cause the processing aspect of the cognitive model to perform differently. For some agents this may cause focus and speed, for others this would cause the error rate to increase. Cyber-FIT version four has a rudimentary concept of error where the agents attempt to restore systems and, depending on their skill level, will do so at a restoral rate coded into the agent ruleset. A more realistic representation of error will take many factors into play such as fatigue, stress, shift length, and others.

Errors can be represented in multiple interesting ways. Defender agents communicate within the model, and in this version all messages are perfectly sent and perfectly received. The processing aspect of the standard model of the mind architecture should process the messages with a possibility of error, like the other errors based on variables such as stress and fatigue. When an agent sends a message, the object representing the message can be incorrect in terms of the terrain agent identified, the details of the vulnerability, indicator of compromise, suspected techniques of the adversary, etc. The defender agents can also make errors when choosing what to work on next, or what tool to use when attempting to gain information. The wrong tool can provide information unrelated to what the question is, or just get to the information at a slower pace.

6.2.2.3 Organizational Modeling

The final aspect of human behavioral modeling which is considered a current limitation and excellent opportunity for improvement is organizational modeling. An excellent way to frame this development effort is to start with what an organization is within an agent-based model. According to Krackhardt and Carley, "Organizations are composed of intelligent adaptive agents who are constrained and enabled by their positions in networks linking agents, knowledge, resources and tasks" [122]. Cyber-FIT agents are connected by directed links which causes them to react. The organization of this version is very sparse and is limiting the veridicality of the model. First and foremost the agents don't have any goals, they just work forever. This could be improved by adding milestones and targets they are working towards in terms of level of security to be achieved. Once a certain level is achieved, the agents will slow down, perhaps simulate a shift change or leveling down of resources needed. This can be an excellent way to simulate spreading

out forces to different areas of the cyber conflict. With a more complex sense of organization, more types of agents can be added as well, with different types of roles which will in turn provide much more variance with how long they spend on different types of tasks.

Another way that the organization can be modeled in terms of shift change would be the amount of information exchanged, and how long the "turnover brief takes". This brings communication costs into play. Any time the agents are communicating, especially when the communication is prolonged (like a turnover brief) the agents are not taking actions on the cyber terrain. The survey I conducted told me that a good estimate of communication time is 30% of total time working. This kind of data provides potential virtual experiments such as how does more efficient communication systems (chat vs. email) impact total information spread and potentially decrease communication costs? It's also clear from speaking with many practitioners that effective leadership is primarily in how they effectively communicate needed and prioritized information to the right team members. This can be modeled message objects that affect how defender agents respond to tasks, correctly report, and interpret expected prioritization, much like a maestro keeps the beat. Information flows around the network by agents communicating which is why the intent is for Cyber-FIT to multi-model with Construct [123].

Another aspect that will be added in a future version is an improved incidence model. One interesting finding with the survey is that cyber security practitioners have very different experiences with resolving incidents. The following figure shows the variance in responses.

Figure 65: Results of question ten from survey

As shown, some think they typically resolve incidents within an hour, some within one day, some within several days, and most longer than several days! This is a perfect type of phenomenon to explore further. There could be many reasons for the differences. First and foremost is definition of incident. Some incidents can take a long time which is completely normal. Imagine a software needs patched, but the manager decides it can wait until the next maintenance window happening over the weekend. So this incident stays open for four additional days. Some organizations might not call that an incident, and only refer to incidents when they are problems that must be remediated as soon as possible. For this definition, we'd expect faster remediation. All in all, a more complex set of behaviors, like those already described in terms of operational behaviors can be adjusted to display a wide variance in time to resolve incidents.

6.2.3 Attacker behavioral modeling

The adversary behavior, or attacker modeling, is also fairly abstract in this version of Cyber-FIT. To review, the basic operations of an attacker agent are: initialize, conduct reconnaissance operations searching for vulnerable systems, match available attacks to those vulnerabilities, deliver the attack payloads, wait for exploitation to occur, and then simulate actions on objectives occurring. This model was validated in several ways with respect to process, and output, but the attacker behavior still has ample room for improvement.

There is not much empirical data on the low level detail of cyber attacks largely due to sensitivity of that data. This means that it is very difficult to create an agent representing a cyber attacker and provide realistic rulesets, in essence we don't know what realistic is. This is a problem that occurs many times in early research and basic models are created, validated, altered, and redeveloped. Cyber-FIT is going through this process on all agent types. Therefore, the question becomes: how best to improve the attacker model? Currently there are two glaring weaknesses: the first is that the attacker can reach into any part of the network and the second is that the attacker is not altering its strategy along the way (it's not very adaptive).

To address the first concern, the attacker should only be able to reach systems relevant to its own specific tactics, techniques, and capabilities at any given time. For instance, when an attacker begins a brand new campaign, it will start by only being able to interact with terrain agents exposed to the internet. This would be simulated edge routers, domain name service servers, web servers, etc. The attacker must learn about specific vulnerabilities and then choose from available techniques that exploit those specific vulnerabilities. The leading industry repository for this type of tactic and technique information is the MITRE ATT&CK database [124]. This database shows the most relevant tactics, with each laying out all of the techniques that can be taken in attempting to exploit using that tactic. Also, rather than a kill chain that must go exactly in order, as

explained using this resource, attacks can move from one tactic to another not necessarily in a prescribed order. For example an attacker might start with reconnaissance, discover some information, move to resource development, then back to reconnaissance to gather more information based on the resources developed. Also, once initial access has happened in the victim network, this iterative process of reconnaissance, resource development, lateral development, further reconnaissance, etc. is a much more realistic model of attacker behavior. With this more complicated process in mind, timing becomes an interesting question. How long do attacks take and how long might an advanced persistent threat be present within a network? Using a more advanced behavioral model will likely mean that the time horizons will be very different calling for longer simulations.

The second concern of adaptability can be addressed by increasing the simulated artificial intelligence of the attacker. In the current model the attacker has no artificial intelligence at all. In an improved model that attacker can have strategies which represent which paths to take given new information gathered from different reconnaissance and resourced development techniques. This would be a more realistic and interesting simulation for the defender agents as well. They could select strategies that are targeted more for preventing initial access rather than hunting APTs (or vice versa) which would lead to differential outcomes based on the resources they prioritize. The attackers could also wait until a sufficient amount of confidence is realized before proceeding with techniques that are likely to leave indicators of compromise. The MITRE database provides examples of what different known cyber adversarial organizations strategies look like. This means that the more advanced adversaries would take one set of strategies while less advanced take a different set of strategies. This leads to an opportunity where the strategies evolve as the attacker agents learn and become aware of opportunities akin to an intelligent tutoring system.

7 References

[1] J. T. Bennett, "Pentagon declares the Internet a war domain," 2011.

[2] H. Zimmerman, "OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection," *IEEE Transactions on Communications,* vol. 28, no. 4, pp. 425-432, 1980.

[3] U.S. Department of Defense, "Joint Publication 3 - 12 Cyberspace Operations," 2018.

[4] Department of Defense, "The Department of Defense Releases the President's Fiscal Year 2023 Defense Budget," 2022.

[5] M. Matishak and L. Seligman, "Biden budget to seek boost to the military's cyber force," *Politico,* 26 May 2021.

[6] J. Johnson, "U.S. government: proposed cyber security spending in FY 2017-2021," Statista, 2021.

[7] Infotechlead, "Top cybersecurity predictions and trends for 2019," *Infotechlead,* 30 December 2018.

[8] Department of Defense, "The DoD Cyber Strategy," Washington D.C., 2015.

[9] M. Boot, War made new: technology, warfare, and the course of history, 1500 to today, Penguin, 2006.

[10] S. Ghamari-Tabrizi, "Simulating the unthinkable: Gaming future war in the 1950s and 1960s," *Social Studies of Science,* vol. 30, no. 2, pp. 163-223, 2000.

[11] M. Gardner, "Mathematical games: The fantastic combinations of John Conway's new solitaire game "life".," *Scientific American,* pp. 120-123, October 1970.

[12] F. Klügl and A. L. Bazzan, "Agent-based modeling and simulation," *AI Magazine,* vol. 33, no. 3, pp. 29-40, 2012.

[13] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems.," *Proceedings of the National Academy of Sciences,* vol. 99, no. 3, pp. 7280-7287, 2002.

[14] B. Heath, R. Hill and C. Frank, "A Survey of Agent-Based Modeling Practices (January 1998 to July 2008)," *Journal Of Artifical Societies and Social Simulation,* vol. 12, no. 4, p. 9, 2009.

[15] D. Welch, G. Conti and J. Marin, "A Framework for an Information Warfare Simulation," in *IEEE Workshop on Information Assurance and Security*, West Point, NY, 2001.

[16] D. L. Bergin, "Cyber-attack and defense simulation framework," *The Journal of Defense Modeling and Simulation,* vol. 12, no. 4, pp. 383-392, 2015.

[17] B. Thompson and J. Morris-King, "An agent-based modeling framework for cybersecurity in mobile tactical networks," *The Journal of Defense Modeling and Simulation,* vol. 15, no. 2, pp. 205-218, 2018.

[18] M. D. Petty, S. E. Barbosa and J. R. Hutt, "Using a mock simulation event and Monte Carlo simulation to compare alternative network architectures for distributed training simulation," *Journal of Defense Modeling and Simulation,* vol. 13, no. 3, pp. 307-320, 2016.

[19] Joint Chiefs of Staff, *Force Readiness Reporting,* Washington D.C.: Joint Staff, 2014.

[20] J. W. Wenger, C. O'Connell and M. C. Lytell, "Retaining the Army's Cyber Expertise," RAND ARROYO CENTER, Santa Monica, CA, 2017.

[21] J. L. Caton, "EXAMINING THE ROLES OF ARMY RESERVE COMPONENT FORCES IN MILITARY CYBERSPACE OPERATIONS," Strategic Studies Institute, US Army War College, Carlisle, PA, 2019.

[22] P. A. Yannakogeorgos and J. P. Geis, II, "The Human Side of Cyber Conflict: Organizing, Training, and Equipping the Air Force Cyber Workforce," Air Force Research Institute, Maxwell AFB, AL, 2016.

[23] J. Miller, "DoD adds another piece to the JWCC puzzle," *Federal News Network,* 7 March 2022.

[24] Joint Staff, "Commander’s Handbook for Assessment Planning and Execution," Department of Defense, Suffolk, VA, 2011.

[25] Joint Staff, "JOINT TRAINING MANUAL FOR THE ARMED FORCES OF THE UNITED STATES," Deparment of Defense, Washington D.C., 2015.

[26] M. Pomerleau, "Army requests $429 million for new cyber training platform," 18 February 2018. [Online]. Available: https://www.fifthdomain.com/dod/2018/02/21/army-requests-429-million-for-new-cyber-training-platform/. [Accessed 5 December 2018].

[27] Alderson Court Reporting, "JOINT HEARING TO RECEIVE TESTIMONY ON THE CYBER OPERATIONAL READINESS OF THE DEPARTMENT OF DEFENSE (OPEN SESSION)," Alderson Court Reporting, Washington D.C., 2018.

[28] Defense Science Board, "Resilient Military Systems and the Advanced Cyber Threat," Department of Defense, 2013.

[29] The White House, "Executive Order on America's Cybesecurity Workforce," 2 May 2019. [Online]. Available: https://www.whitehouse.gov/presidential-actions/executive-order-americas-cybersecurity-workforce/. [Accessed 3 May 2019].

[30] C. M. Macal and M. J. North, "Tutorial on agent-based modeling and simulation," in *Proceedings of the Winter Simulation Conference*, Chicago, 2005.

[31] U.S. Department of Defense, "The Department of Defense Cyber Strategy," 2015.

[32] U.S. Department of Defense Defense Science Board, "Resilient Military Systems and the Advanced Cyber Threat," 2013.

[33] U. Wilensky, "NetLogo," Evanston, IL, 1999.

[34] MITRE, "Common Vulnerabilities Enumeration," MITRE, [Online]. Available: https://cve.mitre.org/. [Accessed 29 March 2022].

[35] Global News Wire, "Military Leaders Take a Closer Look at Innovation Requirements," Global News Wire, 2017.

[36] D. E. Denning, "Framework and principles for active cyber defense," *Computers & Security,* vol. 40, pp. 108-113, 2014.

[37] R. S. Dewar, "The "triptych of cyber security": A classification of active cyber defence," in *6th International Conference On Cyber Conflict (CyCon 2014)*, 2014.

[38] S. Jasper, Strategic cyber deterrence: The active cyber defense option, Rowman & Littlefield, 2017.

[39] E. LeMay, W. Unkenholz, D. Parks, C. Muehrcke, K. Keefe and W. H. Sanders, "Adversary-driven state-based system security evaluation," in *6th International Workshop on Security Measurements and Metrics*, 2010.

[40] M. L. Winterrose and K. M. Carter, "Strategic evolution of adversaries against temporal platform diversity active cyber defenses," 2014.

[41] E. Cayirci and R. Ghergherehchi, "Modeling cyber attacks and their effects on decision process," in *2011 Winter Simulation Conference (WSC)*, 2011.

[42] T. Reed, R. G. Abbott, B. Anderson, K. Nauer and C. Forsythe, "Simulation of workflow and threat characteristics for cyber security incident response teams," in *Human Factors and Ergonomics Society Annual Meeting*, Los Angeles, CA, 2014.

[43] K. E. Heckman, M. J. Walsh, F. J. Stech, T. A. O'boyle and S. R. DiCato, "Active cyber defense with denial and deception: A cyber-wargame experiment," *Computers & Security,* vol. 37, pp. 72-77.

[44] V. Heydari, "Preventing SSH remote attacks using moving target defense (ICCWS), pp.," in *13th International Conference on Cyber Warfare and Security (ICCWS)*, 2018.

[45] S. Moskal, J. Y. Shanchieh and M. E. Kuhl, "Cyber threat assessment via attack scenario simulation using an integrated adversary and network modeling

approach," *The Journal of Defense Modeling and Simulation,* vol. 15, no. 1, pp. 13-29, 2017.

[46] A. Rege, E. Parker , B. Singer and N. Masceri, "A qualitative exploration of adversarial adaptability, group dynamics, and cyber intrusion chains," *Journal of Information Warfare,* vol. 16, no. 3, pp. 1-16, 2018.

[47] M. Cloppert, "Security intelligence: Attacking the cyber kill chain," 2009.

[48] A. Martin, *Military leaders highlight progress in cyber domain during U.S. Senate hearing,* 2018.

[49] V. Delacruz, "Mission Command In and Through Cyberspace: A Primer for Army Commanders," *Military Cyber Defense Review,* 10 December 2015.

[50] C. Onwubiko, "Understanding Cyber Situation Awareness," *International Journal on Cyber Situational Awareness,* vol. 1, no. 1, pp. 11-30, 2016.

[51] P. Barford, M. Dacier, T. G. Dietterich, M. Fredrikson, J. Giffin, S. Jajodia, S. Jha, S. Li, J. Liu, P. Ning and X. Ou, "Cyber SA: Situational awareness for cyber defense," in *Cyber Situational Awarenesss*, Boston, MA: Springer, 2010, pp. 3-13.

[52] Y.-P. Lai and P.-L. Hsia, "Using the Vulnerability Information of Computer Systems to Improve the Network Security," *Computer Communications,* vol. 30, no. 9, p. 2032–2047, 2007.

[53] S. C. Sundaramurthy, S. Bhatt and R. R. Eisenbarth, "Examining intrusion prevention system events from worldwide networks," in *BADGERS '12: Proceedings of the 2012 ACM Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*, 2012.

[54] A. K. Meena, N. Hubballi, Y. Singh, V. Bhatia and K. Franke, "Network Security Systems Log Analysis for Trends and Insights: A Case Study," in *2020 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 2020.

[55] J. Giraldo, D. Urbina, A. Cardenas, J. Valente, M. Faisal, J. Ruths, N. O. Tippenhauer, H. Sandberg and R. Candell, "A Survey of Physics-Based Attack Detection in Cyber-Physical Systems," *ACM Computing Surveys,* vol. 51, no. 4, pp. 1-36, 2018.

[56] G. Longo, "Gaining Cyber Dominance," Carnegie Mellon University Software Engineering Insititue, Pittsburgh, 2015.

[57] M. Pomerleau, "Air Force Squeezes New Cyber Defense Teams Out Of Its Communications Squadrons," 1 October 2021. [Online]. Available: https://www.c4isrnet.com/cyber/2021/10/01/air-force-squeezes-new-cyber-defense-teams-out-of-its-communications-squadrons/. [Accessed 6 January 2022].

[58] M. R. Endsley, "Design and evaluation for situation awareness enhancement," vol. 32, no. 2, pp. 97-101, 1988.

[59] M. R. Endsley and E. S. Connors, "Foundation and Challenges," in *Cyber Defense and Situational Awareness*, Springer, Cham, 2014, pp. 7 - 27.

[60] E. Peters and A. Kitsantas, "The effect of nature of science metacognitive prompts on science students' content and nature of science knowledge, metacognition, and self-regulatory efficacy," *School Science and Mathematics,* vol. 110, no. 8, pp. 382-396, 2010.

[61] A. Poylisher, M. Witkowski, V. D. Veksler, B. E. Hoffman and N. Buchler, "Recording Human Operator Data in Cyber Environments: User Activity Tracker (UAT)," Data and Analysis Center, Aberdeen Proving Ground MD, 2020.

[62] N. Sivasubramaniam, S. J. Liebowitz and C. L. Lackman, "Determinants of new product development team performance: A meta-analytic review," *Journal of Product Innovation Management,* vol. 29, no. 5, pp. 803-820, 2012.

[63] P. Campbell, "Information Assurance (IA) Implementation: A Retrospective," Sandia National Laboratories, Albequerque, NM, 2012.

[64] U.S. Government Accountability Office, "Weapon System Sustainment: Aircraft Mission Capable Rates Generally Did Not Meet Goals and Cost of Sustaining Selected Weapon Systems Varied Widely," U.S. GAO, Washington D.C., 2020.

[65] I. Kotenko and A. Chechulin, "A cyber attack modeling and impact assessment framework," in *5th International Conference on Cyber Conflict (CYCON 2013)*, 2013.

[66] B. J. Wood and R. A. Duggan, "Red teaming of advanced information assurance concepts," in *DARPA Information Survivability Conference and Exposition. DISCEX'00*, 2000.

[67] MITRE, "Indicator Removal on Host," MITRE, [Online]. Available: https://attack.mitre.org/techniques/T1070/. [Accessed 19 May 2021].

[68] N. Altman and K. M. Carley, "ORA User's Guide 2022," Carnegie Mellon University, Pittsburgh, PA, 2022.

[69] M. K. Ahuja and K. M. Carley, "Network structure in virtual organizations," *Organization Science,* vol. 10, no. 6, pp. 741-757, 1999.

[70] M. K. Ahuja, D. F. Galletta and K. M. Carley, "Individual centrality and performance in virtual R&D groups: An empirical study," *Management Science,* vol. 49, no. 1, pp. 21-38, 2003.

[71] G. B. Dobson, T. J. Shimeall and K. M. Carley, "Towards Network Science Enhanced Cyber Situational Awareness," *International Journal on Cyber Situational Awareness,* vol. 1, no. 1, pp. 11-30, 2017.

[72] C. Pellerin, "Cyberspace is the new domain of war," American Forces Press Service, 2010.

[73] S. Ali, S. B. Qaisar, H. Saeed, M. F. Khan, M. Naeem and A. Anpalagan, "Network challenges for cyber physical systems with tiny wireless devices: A case study on reliable pipeline condition monitoring," *Sensors,* vol. 15, no. 4, pp. 7172-7205, 2015.

[74] A. AlDairi and L. Tawalbeh, "Cyber security attacks on smart cities and associated mobile technologies," *Procedia Computer Science,* vol. 109, pp. 1086-1091, 2017.

[75] NATIONAL INITIATIVE FOR CYBERSECURITY CAREERS AND STUDIES, "Workforce Framework for Cybersecurity (NICE Framework)," Cybersecurity and Infrastructure Security Agency.

[76] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington and C. B. Thomas, "Mitre att&ck: Design and philosophy," MITRE, McLean, 2018.

[77] K. M. Carley, "Validating Computational Models," Carnegie Mellon University, Pittsburgh, PA, 1996.

[78] A. M. Householder, G. Wassermann, A. Manion and C. King, "The CERT guide to coordinated vulnerability disclosure," Software Engineering Institute, Pittsburgh, PA, 2017.

[79] M. Zhao, A. Laszka and J. Grossklags, "Devising effective policies for bug-bounty platforms and security vulnerability discovery," *Journal of Information Policy,* vol. 7, no. 1, pp. 372-418, 2017.

[80] H. Homaei and R. S. Hamid , "Seven years of software vulnerabilities: The ebb and flow.," *IEEE Security & Privacy,* vol. 15, no. 1, pp. 58-65, 2017.

[81] B. Miller and D. Rowe, "A Survey SCADA of and Critical Infrastructure Incidents," in *1st Annual Conference on Research in Information Technology*, Calgary, Ablerta, Canada, 2012.

[82] H. Holm, "A Large-Scale Study of the Time Required to Compromise a Computer System," *IEEE Transactions on Dependable and Secure Computing,* vol. 11, no. 1, pp. 2-15, 2014.

[83] C. Simmons, C. Ellis, S. Shiva, D. Dasgupta and Q. Wu, "AVOIDIT: A Cyber Attack Taxonomy," University of Memphis, Memphis, TN, 2009.

[84] C. Meyers, S. Powers and D. Faissol, "Taxonomies of Cyber Adversaries and Attacks: A Survey of Incidents and Approaches," Lawrence Livermore National Lab, Livermore, CA, 2009.

[85] S. Jajodia and S. Noel, "Advanced cyber attack modeling analysis and visualization," George Mason University, Fairfax, VA, 2010.

[86] J. H. Eom , Y. J. Han, S. H. Park and T. M. Chung, "Active cyber attack model for network system's vulnerability assessment," in *International Conference on Information Science and Security*, 2008.

[87] B. Schroeder and G. A. Gibson, "A large-scale study of failures in high-performance computing systems," *IEEE transactions on Dependable and Secure Computing,* vol. 7, no. 4, pp. 337-350, 2009.

[88] H. Holm, "A large-scale study of the time required to compromise a computer system," *IEEE Transactions on Dependable and Secure Computing,* vol. 11, no. 1, pp. 2-15, 2013.

[89] Accenture, "The Cost of Cybercrime," 2019.

[90] D. Dudorov, D. Stupples and M. Newby, "Probability analysis of cyber attack paths against business and commercial enterprise systems," in *European Intelligence and Security Informatics Conference*, 2013.

[91] Pentagon, "Posture statement of Gen. Paul M. Nakasone, commander, U.S. Cyber Command before the 117th Congress," Fort George Meade, MD, 2022.

[92] U.S. Cyber Command Public Affairs, "DOD's Largest Multinational Cyber Exercise Focuses on Collective Defense," Department of Defense, 2021.

[93] M. Pomerleau and A. Eversden, "After years of flat cybersecurity budgets, DoD asks for more money and cyber mission force personnel," C4ISR Net.

[94] L. Giles, The Art of War by Sun Tzu, 2009.

[95] K. Manson, "U.S. Sent Cyber Team to Lithuania Over Russia Hacking Threat," 2022.

[96] Department of the Army, "Field Manual 6-0 Commander and Staff Organization Operations," Washington D.C., 2014.

[97] S. J. Freedberg Jr., "'The Golden 5 Minutes' The Need for Speed in Information Warfare," *Breaking Defense,* p. 2019, 21 October 2019.

[98] G. E. Box, "Robustness in the Strategy of Scientific Model Building," in *Robustness in Statistics*, G. N. W. ROBERT L. LAUNER, Ed., Research Triangle Park, NC, Academic Press, 1979, pp. 201-236.

[99] M. J. North and C. M. Macal, Managing Business Complexity: discovering strategic solutions with agent-based modeling and simulation, New York: Oxford University Press, 2007.

[100] D. B. Johnson, "House set to debate bills on cyber education, President’s Cup and TikTok," *SC Magazine,* 28 February 2022.

[101] US Air Force, "Research Methods and Technologies for Blended Live and Synthetic Personalized Learning, Modeling and Assessment Open Broad Agency Announcement," 2020.

[102] W. Newhouse, S. Keith, B. Scribner and G. Witte, "National initiative for cybersecurity education (NICE) cybersecurity workforce framework," Cybersecurity and Infrastrcuture Security Agency, 2017.

[103] K. M. Carley, "Simulating society: The tension between transparency and veridicality," in *Agents*, Chicago, IL, 2002.

[104] G. B. Dobson, A. Rege and K. M. Carley, "Informing active cyber defence with realistic adversarial behaviour," *Journal of Information Warfare,* vol. 17, no. 2, pp. 16-31, 2018.

[105] L. Bilge and T. Dumitras, "Before we knew it: an empirical study of zero-day attacks in the real world," in *ACM conference on Computer and communications security*, Raleigh, NC, 2012.

[106] Applied Computer Research Inc., "Identifying IT Markets and Market Size by Number of Servers," 2011. [Online]. Available: https://www.missioncriticalmagazine.com/ext/resources/MC/Home/Files/PDFs/WP_ACR-IT-Server-Market.pdf. [Accessed 29 March 2022].

[107] G. B. Dobson, T. G. Podnar, A. D. Cerini and L. J. Osterritter, "R-EACTR: A Framework for Designing Realistic Cyber Warfare Exercises," Software Engineering Institute, Pittsburgh, PA, 2017.

[108] T. G. Podnar, G. B. Dobson, D. D. Updyke and W. E. Reed, "Foundation of Cyber Ranges.," Software Engineering Institute, Pittsburgh, PA, 2021.

[109] C. Schreiber and K. Carley, "Going beyond the data: Empirical validation leading to grounded theory," *Computational & Mathematical Organization Theory,* vol. 10, no. 2, pp. 155-164, 2004.

[110] G. B. Dobson, T. J. Shimeall and K. M. Carley, "Towards Network Science Enhanced Cyber Situational Awareness," *International Journal of Cyber Situational Awareness,* vol. 2, no. 1, pp. 11-30, 2017.

[111] K. M. Carley, "Computational and mathematical organization theory: Perspective and directions," *Computational & mathematical organization theory,* vol. 1, no. 1, pp. 39-56, 1995.

[112] S. Gordon and R. Ford, "On the definition and classification of cybercrime," *Journal in computer virology,* vol. 2, no. 1, pp. 13-20, 2006.

[113] E. LeMay, W. Unkenholz, D. Parks, C. Muehrcke, K. Keefe and W. H. Sanders, "Adversary-driven state-based system security evaluation," in *6th International Workshop on Security Measurements and Metrics*, 2010.

[114] M. R. Endsley, "Measurement of situation awareness in dynamic systems," *Human factors,* vol. 37, no. 1, pp. 65-84, 1995.

[115] M. T. Brannick and C. Prince, "An overview of team performance measurement," in *Team performance assessment and measurement*, New York, Psychology Press, 1997, pp. 15-28.

[116] G. Hallam and D. Campbell, "The measurement of team performance with a standardized survey," in *Team performance assessment and measurement*, New York, Psychology Press, 1997, pp. 167-184.

[117] M. Granåsen and D. Andersson, "Measuring team effectiveness in cyber-defense exercises: a cross-disciplinary case study," *Cognition, Technology & Work,* vol. 18, no. 1, pp. 121-143, 2016.

[118] K. M. Carley and V. Hill, "Structural change and learning within organizations," *Dynamics of organizations: Computational modeling and organizational theories,* pp. 63-92, 2001.

[119] Canada National Defence, "DND/CAF Sweeps the Podium at U.S. Cyber Command “Cyber Flag 21-2”".

[120] S. Wise, "Can a team have too much cohesion? The dark side to network density," *European Management Journal,* vol. 32, no. 5, pp. 703-711, 2014.

[121] J. E. Laird, C. Lebiere and P. S. Rosenbloom, "A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics," *Ai Magazine,* vol. 38, no. 4, pp. 13-26, 2017.

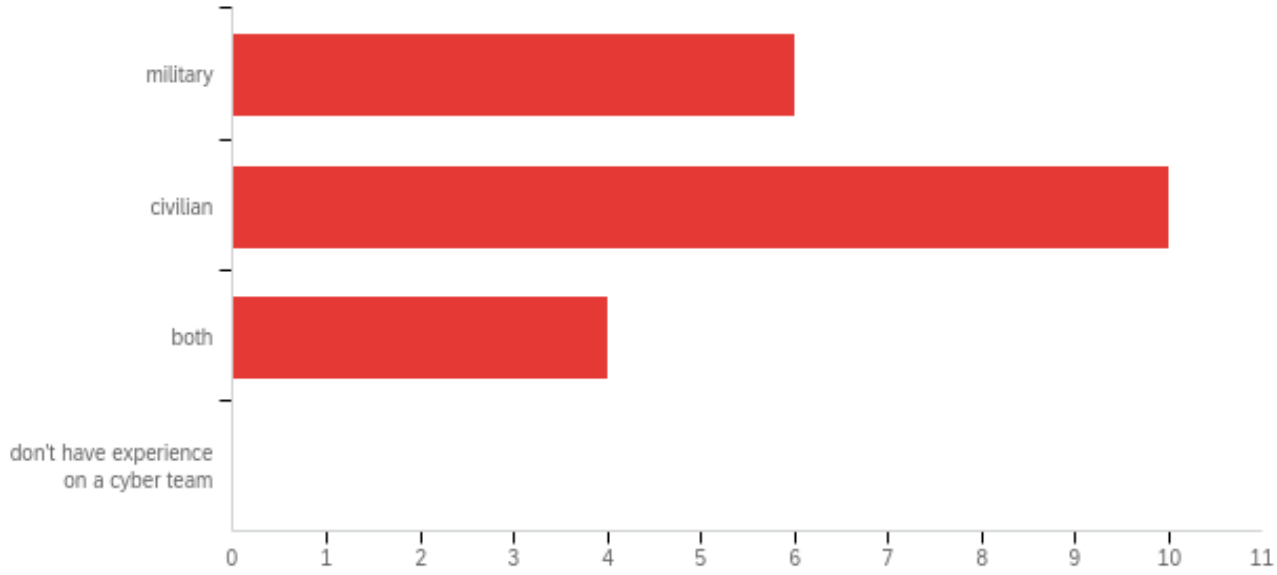
[122] D. Krackhardt and K. Carley, "A PCANS model of structure in organizations," in *Proceedings of the International Symposium on Command and Control Research and Technology, Evidence Based Research*, Vienna, VA, 1997.

[123] C. Schreiber, S. Singh and K. M. Carley, "Construct-a multi-agent network model for the co-evolution of agents and socio-cultural environments," Carnegie Mellon University, Pittsburgh, PA, 2004.

[124] MITRE, "ATT&CK," MITRE, [Online]. Available: https://attack.mitre.org/. [Accessed 28 June 2022].

7.1 Appendix A – Survey Results

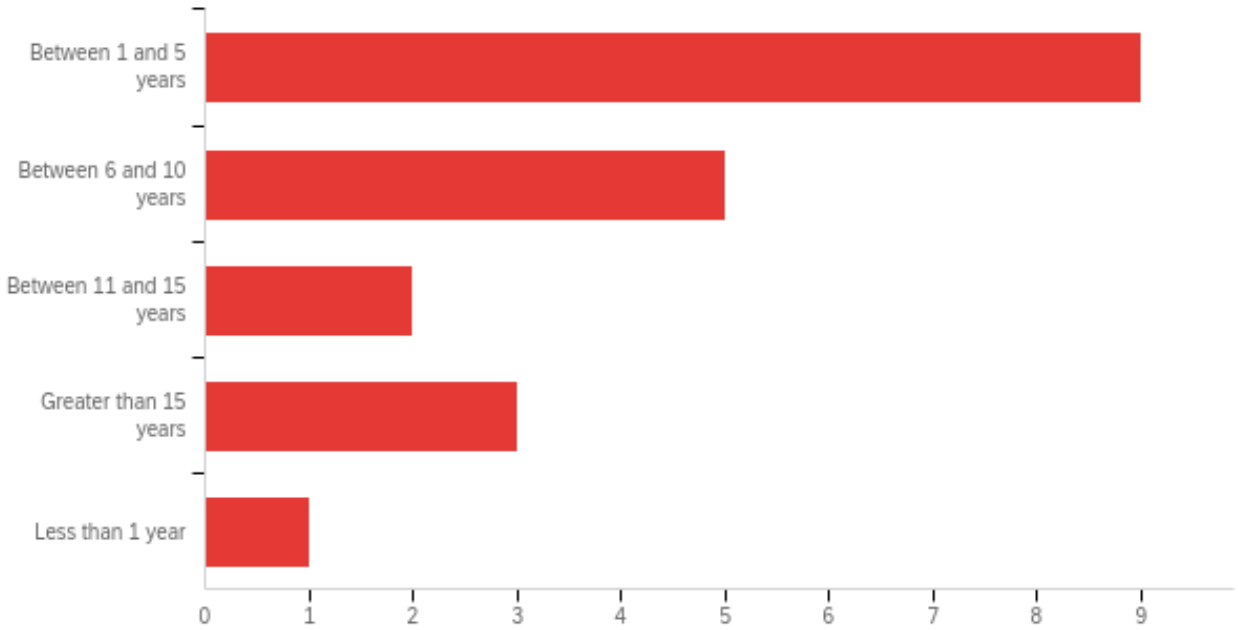
Q1 - Do you have experience on a military or civilian cyber team?



#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Do you have experience on a military or civilian cyber team?	1.00	3.00	1.90	0.70	0.49	20

#	Answer	%	Count
1	military	30.00%	6
2	civilian	50.00%	10
3	both	20.00%	4
4	I don't have experience on a cyber team	0.00%	0
	Total	100%	20

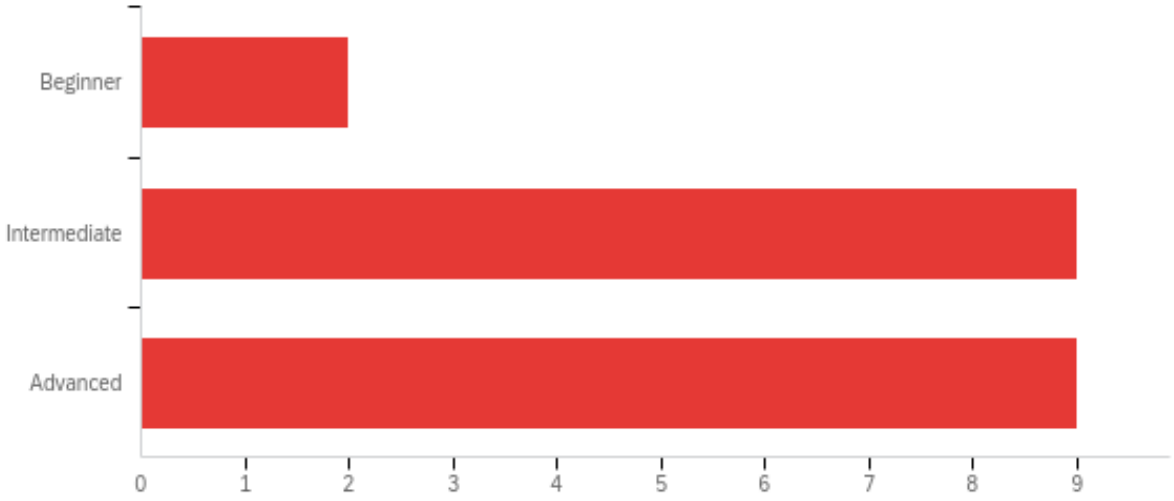
Q2 - How many years of experience do you have on a cyber security team?



#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	How many years of experience do you have on a cyber security team?	1.00	5.00	2.10	1.26	1.59	20

#	Answer	%	Count
1	Between 1 and 5 years	45.00%	9
2	Between 6 and 10 years	25.00%	5
3	Between 11 and 15 years	10.00%	2
4	Greater than 15 years	15.00%	3
5	Less than 1 year	5.00%	1
	Total	100%	20

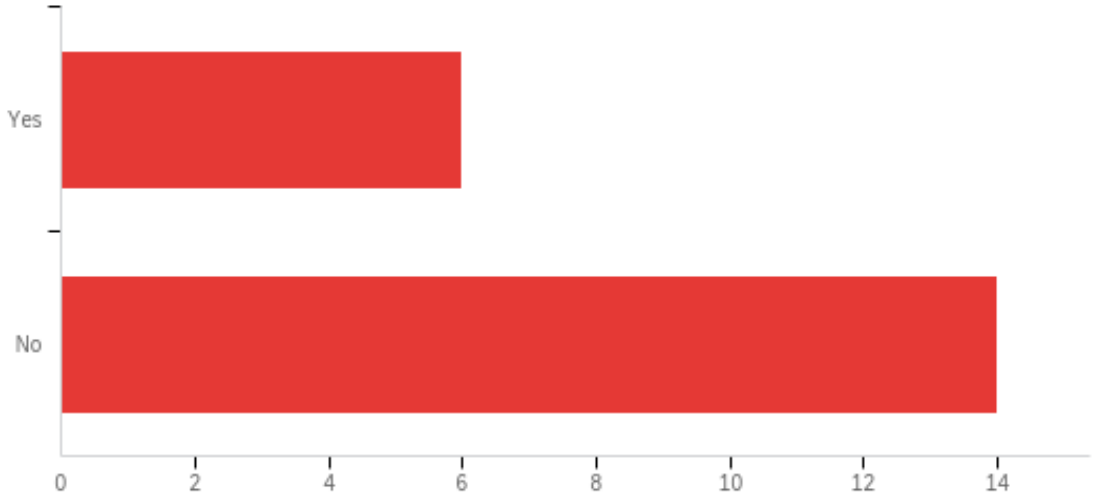
Q3 - What is your assessment of your current cyber security skill level? (This is your assessment of the technical cyber security skills needed to complete tasks associated with your job)



#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	What is your assessment of your current cyber security skill level? (This is your assessment of the technical cyber security skills needed to complete tasks associated with your job)	1.00	3.00	2.35	0.65	0.43	20

#		Answer	%	Count
1		Beginner	10.00%	2
2		Intermediate	45.00%	9
3		Advanced	45.00%	9
		Total	100%	20

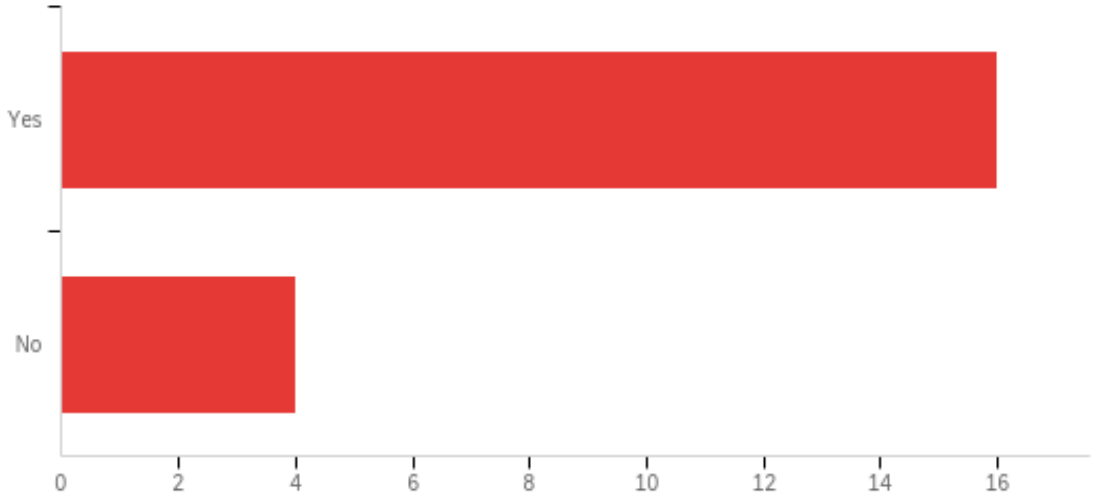
Q4 - Do you have an associate's degree in information technology, computers, or cyber security?



#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Do you have an Associate's degree in information technology, computers, or cyber security?	1.00	2.00	1.70	0.46	0.21	20

#	Answer	%	Count
1	Yes	30.00%	6
2	No	70.00%	14
	Total	100%	20

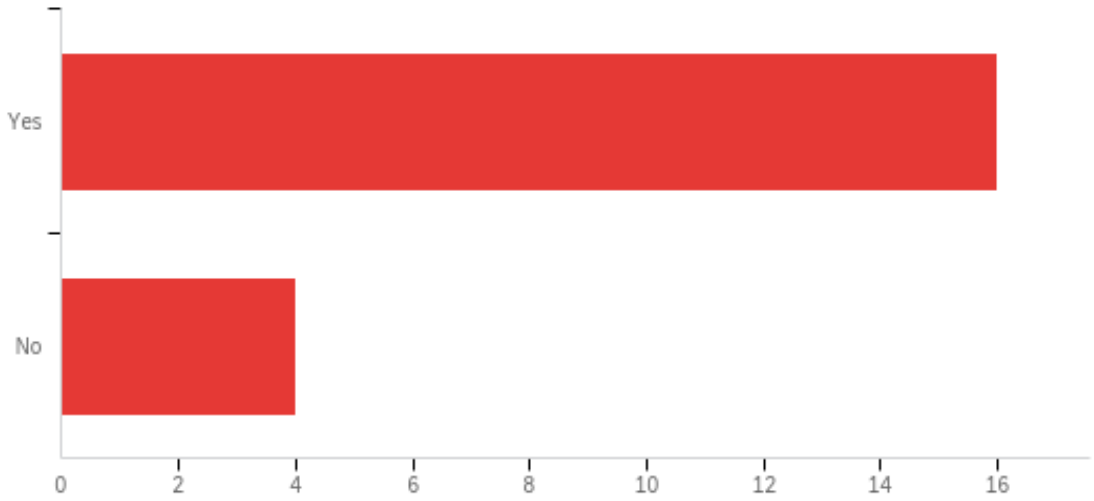
Q5 - Do you have a bachelor's degree in information technology, computers, or cyber security?



#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Do you have a Bachelor's degree in information technology, computers, or cyber security?	1.00	2.00	1.20	0.40	0.16	20

#	Answer	%	Count
1	Yes	80.00%	16
2	No	20.00%	4
	Total	100%	20

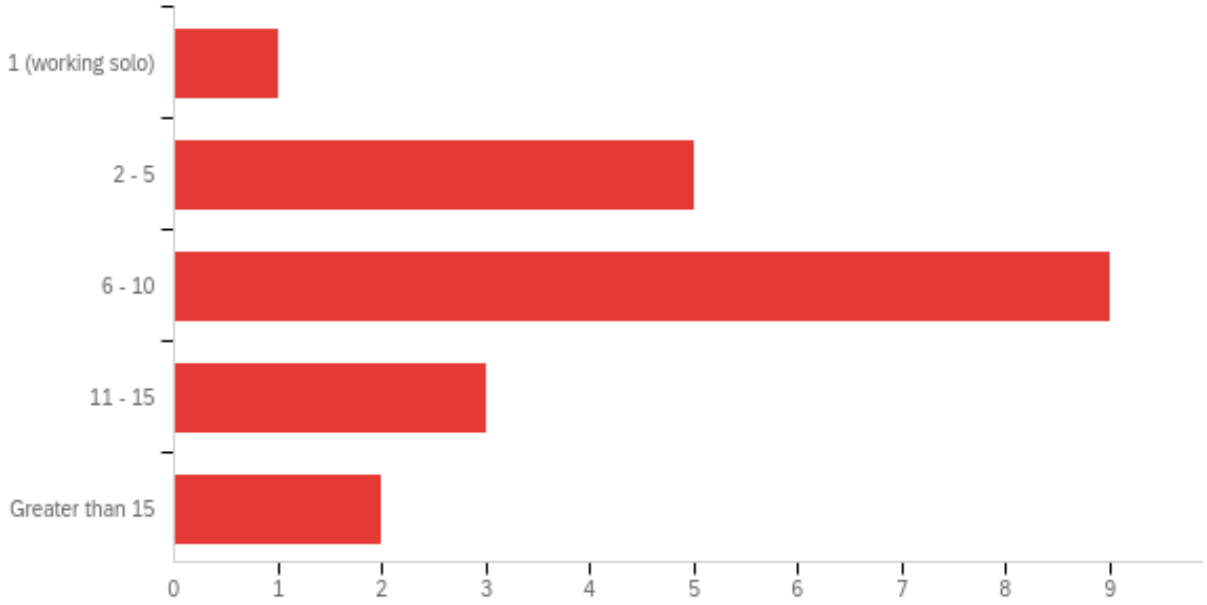
Q6 - Do you have an industry recognized certification such as Security+, CISSP, or other?



#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Do you have an industry recognized certification such as Security+, CISSP, or other?	1.00	2.00	1.20	0.40	0.16	20

#	Answer	%	Count
1	Yes	80.00%	16
2	No	20.00%	4
	Total	100%	20

Q7 - How many personnel are typically on cyber teams that you've worked on?



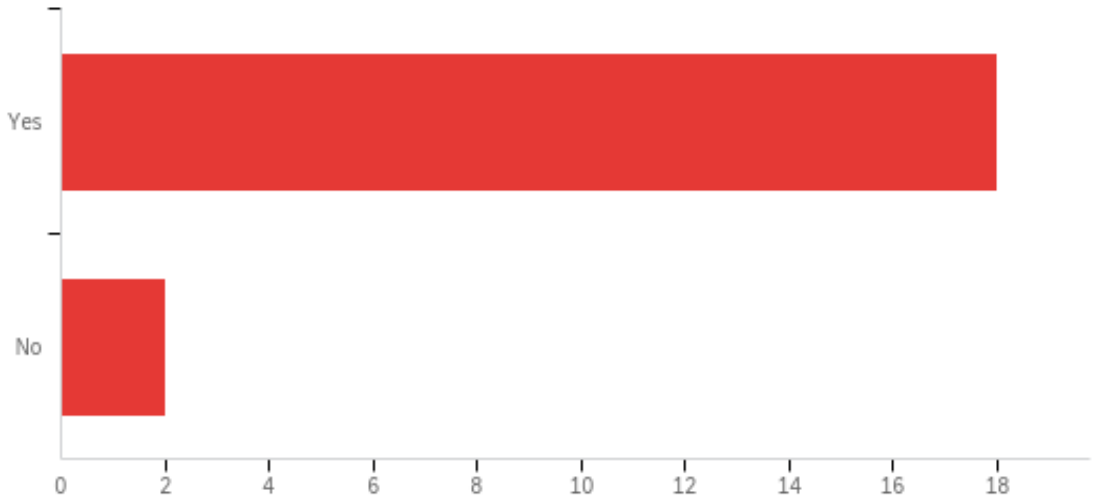
#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	How many personnel are typically on cyber teams that you've worked on?	1.00	5.00	3.00	1.00	1.00	20

#	Answer	%	Count
1	1 (working solo)	5.00%	1
2	2 - 5	25.00%	5
3	6 - 10	45.00%	9
4	11 - 15	15.00%	3
5	Greater than 15	10.00%	2
	Total	100%	20

Q8 - During normal team operations, what percentage of the time are you doing the following types of tasks? (Your answers must add up to 100, shown in the total)

#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Surveying systems	0.00	90.00	20.25	18.47	341.19	20
2	Updating and working on systems	0.00	60.00	19.25	14.08	198.19	20
3	Interacting with other team members	0.00	60.00	29.00	14.46	209.00	20
4	Reporting about systems	0.00	50.00	20.25	11.67	136.19	20
5	Other	0.00	45.00	11.25	12.54	157.19	20

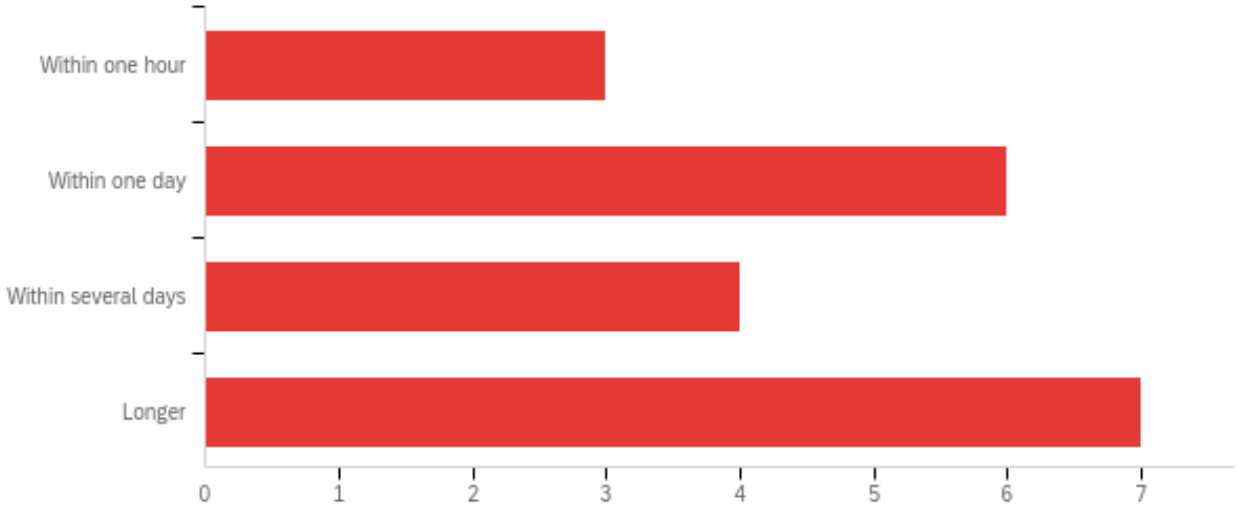
Q9 - Have you experienced cyber incidents in an operational environment?



#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Have you experienced cyber incidents in an operational environment?	1.00	2.00	1.10	0.30	0.09	20

#	Answer	%	Count
1	Yes	90.00%	18
2	No	10.00%	2
	Total	100%	20

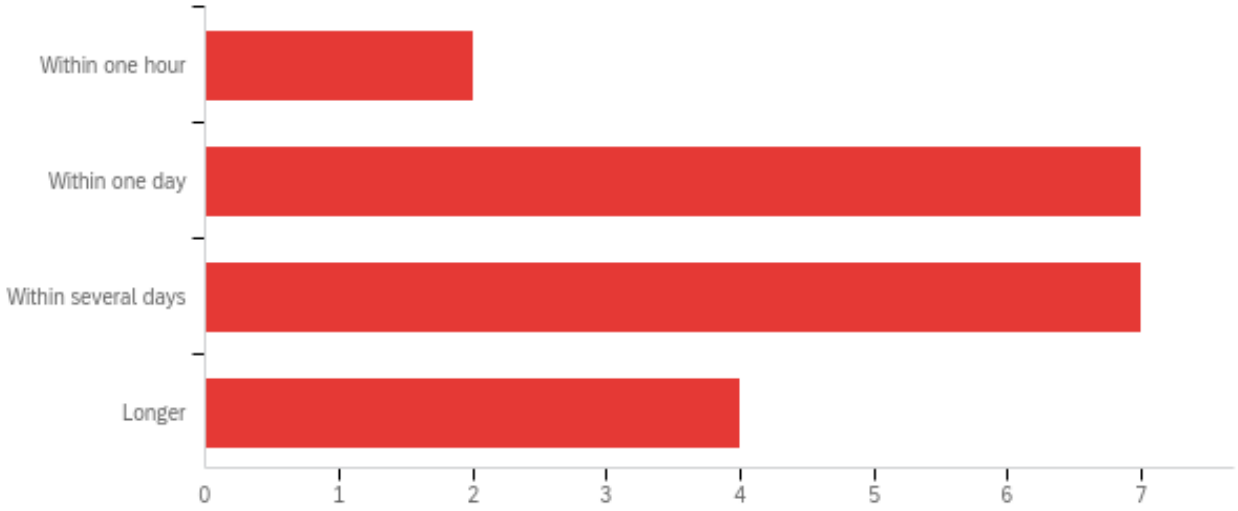
Q10 - When experiencing cyber incidents, how long after the incident actually began, on average, are you alerted (through a security system or human investigation)



#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	When experiencing cyber incidents, how long after the incident actually began, on average, are you alerted (through a security system or human investigation)	1.00	4.00	2.75	1.09	1.19	20

#		Answer	%	Count
1		Within one hour	15.00%	3
2		Within one day	30.00%	6
3		Within several days	20.00%	4
4		Longer	35.00%	7
		Total	100%	20

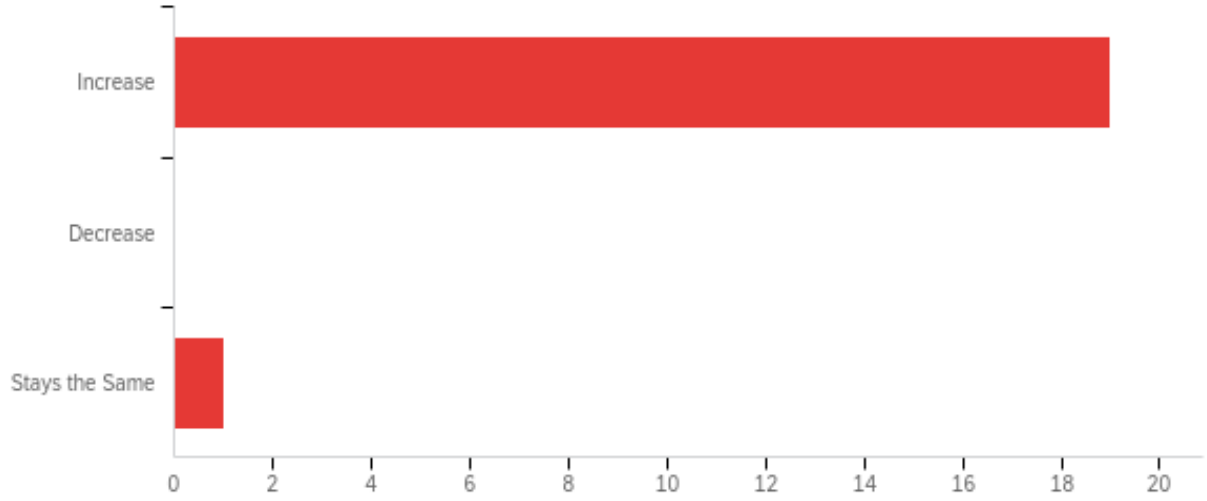
Q11 - When experiencing cyber incidents, how long after the incident is identified, on average, does it take to mitigate?



#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	When experiencing cyber incidents, how long after the incident is identified, on average, does it take to mitigate?	1.00	4.00	2.65	0.91	0.83	20

#	Answer	%	Count
1	Within one hour	10.00%	2
2	Within one day	35.00%	7
3	Within several days	35.00%	7
4	Longer	20.00%	4
	Total	100%	20

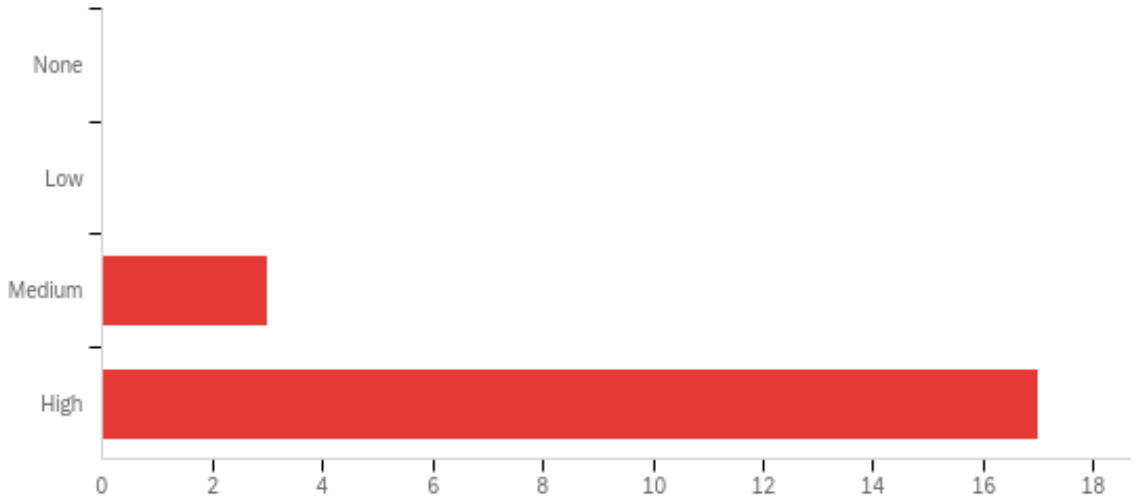
Q12 - Think about how much you interact with other members of your cyber security team. When comparing the amount of interaction you have, does the level of interaction during an incident, as compared to normal operations:



#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Think about how much you interact with other members of your cyber security team. When comparing the amount of interaction you have, does the level of interaction during an incident, as compared to normal operations:	1.00	3.00	1.10	0.44	0.19	20

#	Answer	%	Count
1	Increase	95.00%	19
2	Decrease	0.00%	0
3	Stays the Same	5.00%	1
	Total	100%	20

Q13 - During a cyber incident, is the level of interaction within your cyber team



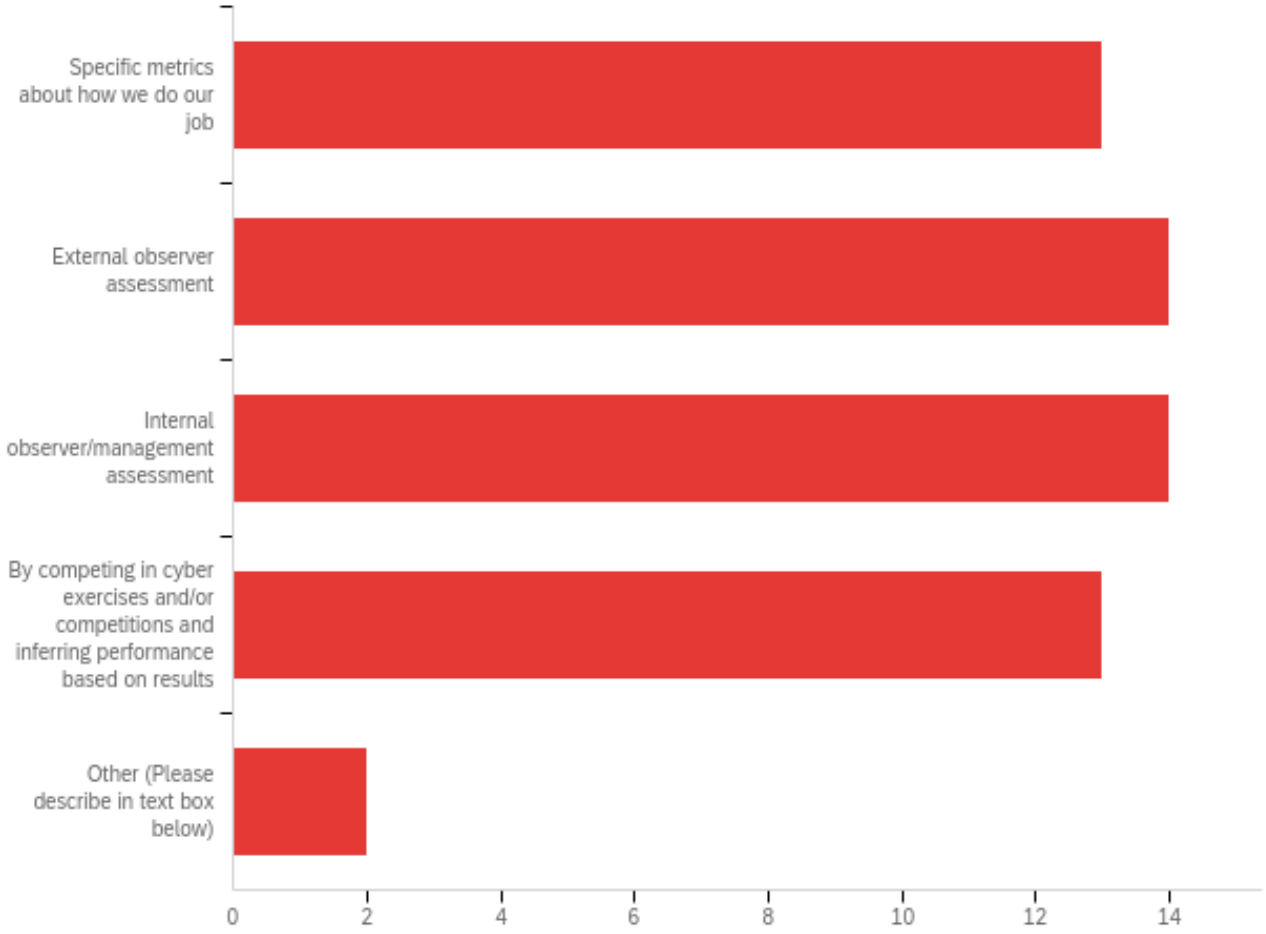
#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	During a cyber incident, is the level of interaction within your cyber team	3.00	4.00	3.85	0.36	0.13	20

#	Answer	%	Count
1	None	0.00%	0
2	Low	0.00%	0
3	Medium	15.00%	3
4	High	85.00%	17
	Total	100%	20

Q14 - During operations where an incident has been recognized, what percentage of the time are you doing the following types of tasks? (Your answers must add up to 100, shown in the total)

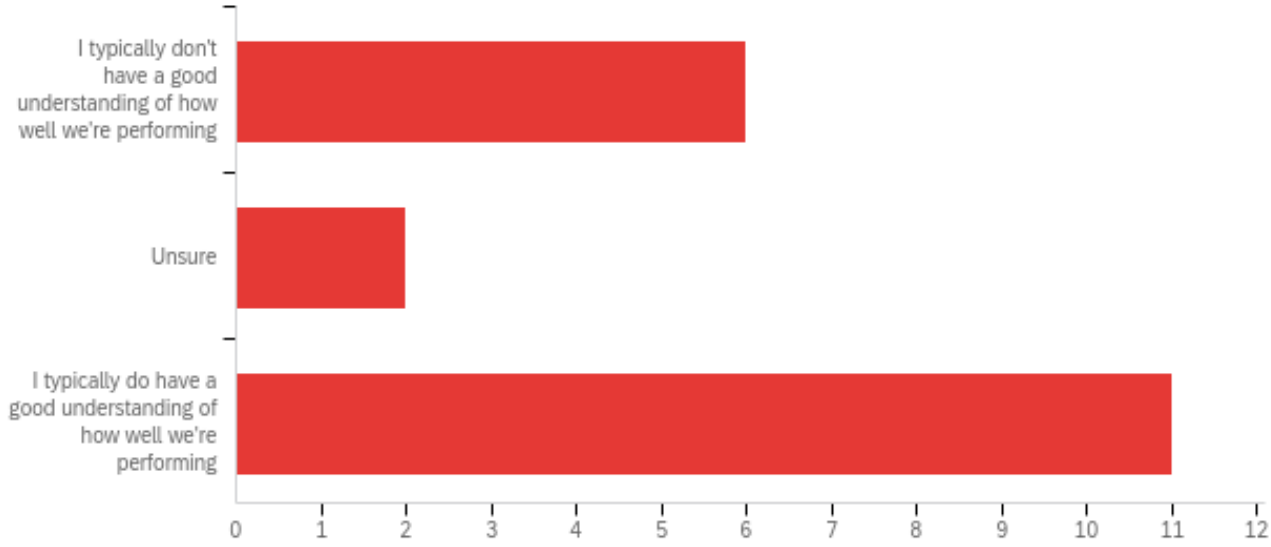
#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Surveying systems	0.00	50.00	19.00	14.20	201.50	20
2	Updating systems	0.00	40.00	10.50	12.34	152.25	20
3	Interacting with team members	10.00	50.00	36.00	11.58	134.00	20
4	Reporting about systems	0.00	50.00	17.75	14.18	201.19	20
5	Restoring systems that have been compromised	0.00	50.00	11.75	12.07	145.69	20
6	Other	0.00	30.00	5.56	8.48	71.91	18

Q15 - On cyber teams I've been on, our performance has been assessed in the following ways: (select all that apply)



#	Answer	%	Count
1	Specific metrics about how we do our job	23.21%	13
2	External observer assessment	25.00%	14
3	Internal observer/management assessment	25.00%	14
4	By competing in cyber exercises and/or competitions and inferring performance based on results	23.21%	13
5	Other (Please describe in text box below)	3.57%	2
	Total	100%	56

Q16 - On cyber teams you've been on, do you typically have an understanding of how well the team is performing?



#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	On cyber teams you've been on, do you typically have an understanding of how well the team is performing?	1.00	4.00	2.95	1.36	1.84	19

#	Answer	%	Count
1	I typically don't have a good understanding of how well we're performing	31.58%	6
3	Unsure	10.53%	2
4	I typically do have a good understanding of how well we're performing	57.89%	11
	Total	100%	19

7.2 Appendix B – Focus Group Details

Part 1 Review of Survey

1. Question 13: 19/20 say interaction increases during an incident.
    a. What about pressure, stress?
        i. Stress depends on environment ie know the environment, cyber key terrain , etc stress not as high due to familiarity. Less familiar will see a more rapid burst of communications.
        ii. Stress will increase with going down rabbit hole effect, ie – trying to search but not finding. Also, with "reaction" mode.
        iii. Stress will be dependent on reporting requirements – so in a response plan will have cadence that tells when to update/how to deal with incident manager. This could be simulated by team lead agent communications. *need to look at pirate combat white paper. Also, business rules will affect stress of team. Ie – cant take down stock trading machines on Friday at 4:00 PM. Could provide cadence schedule to the agents on when they're due to report. Incident management plan would have roles for team members based on expertise/experience. Knowledge of where the adversary may or may not be induces stress.
2. Question 22: Only 11 out of 19 typically understand how well the team is performing
    a. Why is this?
        i. Performance will be dependent on tool set familiarity. And must have an actual event of some sort to define performance. Typically a mixed bag. Less experience typically less knowledge of how to do the things necessary to perform well.
        ii. Likely has element of silo. Example – I do my job but not sure how the collective is performing, therefore the incident manager has onus through experience to tie this together. This means that leadership very important.
        iii. Table tops good for providing SA on what it means for team to perform.

Part 2 Validation of Cyber-FIT

Questions to discuss as a group and document feedback:

1. How is cyber team performance currently measured?
    a. Bring in purple team to assess and receive feedback from external assessor. Very hard to define metrics. Have a team member run a tool that an expert

knows well, and compare/contrast to the expert use of the tool. Possible examples is how much malware is found on network. Not very useful. More useful is to have team do red teaming on own organization.

b. Attrition rate makes it very difficult to mature the metrics and measures. So, you are constantly grading a different team.
c. Environment is changing, so similar to attrition, you are not measuring the same playing field
d. Most metrics are a measure of how the tools are performing.

2. How could cyber team performance be measured better, given no constraints?
   a. Use case management data. Ie – what task needed to perform and how quickly to close out. What needed to happen to close out a case. Opportunity to use case management data for analytics.
   b. Would do it from user end of the IT. Post incident are we better off based on cyber team work?
   c. Simultaneously run two teams through an event see which is better.
   d. Ideally if you have blank check virtualize the environment and red team on it forever.
   e. Do it from type of attack perspective to see who responds best to different problems.
3. Walk through each of the Cyber-FIT performance measures for feedback
   a. Of these metrics which are most helpful and already used?
      i. Time to compromise is one of most used – ie purple team events the bosses always want to know how hard to get to crown jewles.
      ii. Time to detect is one of most used and a focus of how to use a CPT toolset.
      iii. Time to restore is one of most critical so, again from biz perspective.
      iv. All of the other measures look great, hard to implement, but definitely moving in right direction
4. Demo video of Cyber-FIT
   a. How can software like this be used for DoD research, war-gaming, training and cyber team modelling?
      i. Cyber Warfare Publication has 4 core functions. A tool could predict how long different things should take.
      ii. Identify training gaps ie if more people added how does team change versus add training for specific skills
      iii. Use NIST Cyber Framework and determine if these skills/tasks are valid because problem is a lot are hard to differentiate, determine when one is doing one and not another, etc.
      iv. War-gaming for training and conflict course of action analysis