

System Engineering a Scalable Cyber Warfare Test Environment

Ke-Thia Yao, John J. Tran, Dan M. Davis, Daniel P. Burns & Robert F. Lucas
Information Sciences Institute and Institute for Creative Technologies
University of Southern California

The Test and Evaluation (T&E) community is at the forefront of meeting the challenges presented by the cyber security attacks on U.S. infrastructure, industry and individuals. The Information Sciences Institute of the University of Southern California, along with personnel from the California Air National Guard and a major National Laboratory, have developed a set of simulations and techniques for use by small cyber security units. These new approaches should be directly applicable to the T&E. This paper focuses on our research and development (R&D) efforts, which used HPC to stand up this low-cost fully operable simulated cyberspace environment. The designers recognized the necessity of having effective methods of capturing, logging, archiving and displaying the resultant data. The paper will close with an analysis of the utility of this approach, its extensibility into other areas, and future research requirements.

Key Words: System Engineering, Cyber Security, Scalability, Small Units,

Two challenges are consistently in the news at this time: ensuring cyber security and managing massive data sets. This paper addresses the authors' approach to responding to the confluence of these two problem areas. Both challenges have been extensively documented and effectively articulated. Both have attracted significant interest and research focus from the defense establishment, society at large and academia. The rapid explosion of information technology is capable of generating virtually inconceivable torrents of information, making it difficult for users to recognize, understand and incorporate available responses to problems, both those that are extant and those that are emerging. This paper presents a set of technologies, skills and insights garnered from the operation of very large and continentally dispersed simulation networks involving hundreds of participants and thousands of CPU's. The lessons learned from these efforts have been effectively instantiated in a recent effort standing up a

cyber security exercise. That exercise was designed to be scalable from a single participant to a globally-connected event involving thousands. The initial raison d'être and focus was providing an easily configured cyber security exercise program for use by small active duty and reserve units, which are typically made up of from ten to thirty personnel. They would be supported by scalable computing enabling them by operating over the range of a few local personal computers up to tens of thousands of nodes on major high performance Linux clusters operated by the High Performance Computing Modernization Program. Even the small unit exercises will generate vast amounts of data and it is the team's observation that global exercises have in the past generated, and are anticipated to in the future to generate, astronomical volumes of data. Meeting that challenge has required the innovative use of superior techniques and technologies.

The hopes spawned by the availability of new technologies are often dashed by one of two results: 1) the technologies fail to live up to their performance promises and are eclipsed by improvements in existing technologies or 2) implementation of the technologies is so cumbersome and the maintenance of the new code is so onerous as to render the modest improvement meaningless. This raises the meta-issue of how to assess new technologies and how to implement them efficaciously. Again the authors offer their insights into an approach grounded in a comprehensive understanding of the needs and procedures of the intended users to define the parameters and quantify the requirements of the analytic and data management systems. They will discuss how this was accomplished in the design and coding of the system for the small unit cyber exercises: Cyber Quick-Reaction Training Environment (CQRTE – pronounced “kwerty,” like the keyboard). In doing so, they were able to avoid both of the disappointing results alluded to above.

The paper will close with a discussion of the utility of this approach, a primer on its implementation, and some speculation as to its extensibility for other uses and for other purposes.

Prior Art Developed on Earlier Projects

High Performance Computing has shown the ability to enable large-scale battlefield simulations to scale up the ten million semi-automated forces entities¹. Using HPC instead of tens of thousands of live participants has a number of benefits ranging from cost savings all the way to the ease of automated logging of entity activities, location, and status.

Many of the designers of the CQRTE system and some of the authors on this paper developed their approaches to data management

and distributed computing in the Joint Experimentation on Scalable Parallel Processors project². That was a trans-continently distributed battlespace simulation that required a scalable logging capability. This work was enabled by using thousands of nodes of a high-performance Linux cluster to produce large amounts of data recording the actions of millions of independent agent entities representing forces, vehicles, terrain and civilian populations. Managing that torrent of data became very problematic³.

The JESPP system and the distribution of the computers and the human-in-the-loop participants are shown in Figure 1, below.

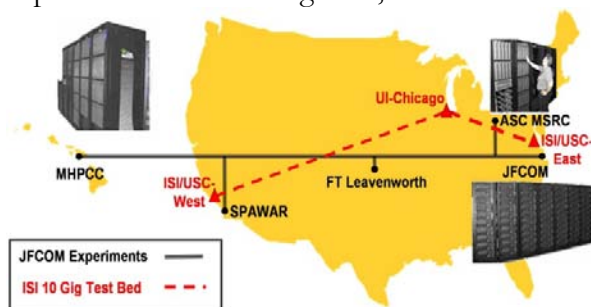


Figure 1. JFCOM Experimentation System

For most of the JESPP project, interest-limited message exchange was done using ISI’s MeshRouter design, as its scalability was far better than other designs.

Data Management Design for JESPP Project

The analysts in the JESPP project were interested in how well higher level mission tasks and objects were satisfied. A Measure of Effectiveness (MOE) is a question or measure, designed to show how well particular tasks were satisfied with respect to a system⁴. A Measure of Performance (MOP) is typically quantitative measure of a system characteristic used to support a MOE, Figure 2: Dimensions of Interest and Measures of Performance. For example, sample MOE questions

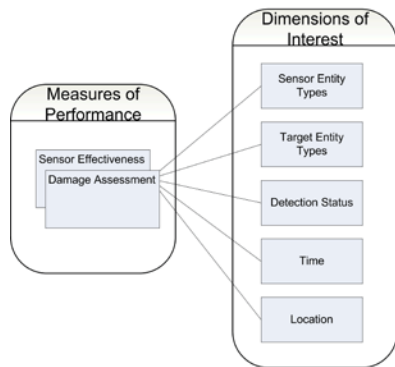


Figure 2: Dimensions of Interest and Measures of Performance

were “Could the red forces be pinned?”, or “Could the sensors detect red force movement within urban environment?”. MOPs supporting these MOEs typically included parameters like: percentage of red forces killed/damaged, percentage of blue forces kill/damaged, time take to cross terrain, percentage of forces detected within sensor footprint, percent of forces detected total, and percentage of detection by sensor type by terrain type by time of day.

Analysis Data Model

A principle mandate of JESPP was to study the effectiveness of future Intelligence, Surveillance and Reconnaissance (ISR) sensors in helping “Blue Forces” operate in complex urban environments. The Sensor/Target Scoreboard provides a visual way of quickly comparing the relative effectiveness of individual sensor platforms and sensor modes against different types of targets. Sensor/Target Scoreboard is a specific instance of the more general multidimensional analysis⁵). We used the Sensor/Target Scoreboard to motivate the discussion. In a 2005 I/ITSEC paper, the data management and analysis tool Scalable Data Grid, which uses multidimensional analysis, is fully described⁶.

The Analysis Data Model (ADM) is defined in terms of multidimensional analysis. Two key

concepts in the ADM are dimensions of interest and Measures of Performance (Figure 2). For large simulations, the magnitude of data collected ranged in the terabytes. Dimensions categorize and partition the data along lines of interest to the analysts. Defining multiple crosscutting dimensions aided in breaking the data into smaller orthogonal subsets associated measurement units. Choosing the granularity of these units aided in determining the size of the subsets. The targets were often grouped together, for example, by transportation mode: air, ground and sea.

Hierarchical dimensional units were also possible. For example, the analysts may have wanted to subdivide the sensor platform category into the sensor modes: MTI (moving target indicators), SAR (synthetic aperture radar), images, video, and acoustic. Multiple unit decompositions of the same dimension are allowed.

In the case of the sensor/target scoreboard, the MOP was an integer count of the number of times a sensor has detected a target. The aggregation operator is the addition operator. Sometimes mean and variance performance measures were of interest to analysts. For example, instead of integer detection counts, the sensor/target could have been extended to maintain a floating point number indicating degree of uncertainty. Then, in that case it made sense to measure the mean and the variance of uncertainty. Mean and variance operators satisfied the associative property. These measures were directly computable from associative measure. Mean was computable from two associative measures: the count of number of detections (or uncertainty), and the sum of uncertainty. Variance required an additional sum of squares of uncertainty. Let X be the uncertainty, and n be the count of number of detections:

$$\text{Mean} = \frac{\sum X}{n}; \quad \text{Var} = \frac{n\sum X^2 - (\sum X)^2}{n^2}$$

Typically MOE decomposed into multiple performance measures. If it was possible to decompose these measures along the same dimensions, or as they are sometimes called, conforming dimensions, then it was possible to compare these measures. The type of question analysts might have asked was “How more likely are damages to enemy target entities if they have been detected by sensors?”. Let X be sensor effectiveness and Y be damage assessment. If an additional measure was defined that was the sum of X times Y , then we could determine covariance between damage and detection by using the well-known “mean” operator:

$$\text{Cov}(X, Y) = \text{Mean}(XY) - \text{Mean}(X)\text{Mean}(Y)$$

Logging Data Model

The Logging Data Model (LDM) described the content and format of data being logged. The system used relational databases to store the logged data. The ISI LDM was automatically generated from the simulation description document that described classes and interactions, having attributes and parameters respectively. Attributes are mapped to multiple columns, to rows in a sub-table or to multiple columns in the top-level table. Complex data types with high or unbounded cardinality, such as arrays, are mapped to sub-tables. The sub-table contains keys referencing the parent table, sequence column indicating the order in the array and data columns representing the actual data. Interactions and their corresponding parameters are handled similarly to the object classes and their attributes. The purpose of the relational schema is for the efficient storage of log data intercepted during the federation execution. It has been pointed

out that the primary purpose of relational schema is data integrity.

Distributed Implementations and Results

To work in distributed environments an additional layer was needed to define on top to aggregate multidimensional cubes distributed across different machines. The left-hand side of Figure 3 below depicts a single three-dimensional sensor/target/detection status score-cube. It represents only a partial, incomplete view. To generate a complete view, cubes from other simulation federates have to be aggregated. The right-hand side of Figure 3 depicts a tree summing together all the distributed cubes. Again, the associative and commutative properties of the aggregation operator were used, while the raw data was not sent.

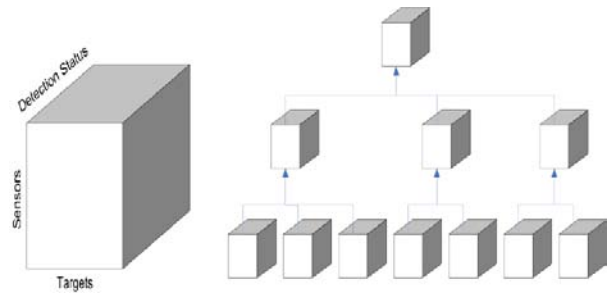


Figure 3 Distributed Data Analysis

Analysis of Data Management

The ability to capture and log detail message traffic from very large scale simulations exceeded the ability of humans to analyze and comprehend that data. A framework for quickly translating these operational-level log data into analyst-level data was implemented. The framework explicitly defined a two-level data model that separated the operational logging data model from the analysis data model. The agility of the framework resulted from being able to isolate changes to the logging data model as a result of changes to the simu-

lation parameters, and from being able to quickly define analysis data models that matched analysts' notion of MOEs and MOPs. The secret to a good data management system was to sit with the analysts during a series of simulation to more fully understand what their needs and their constraints were. Only then was the system designed.

Development of a Small Unit Data Management System

In this section, the conception, design and implementation of a data management system for a small unit cyberwarfare simulator will be detailed and analyzed.

The Cyber Security Threat

The President has identified cyber security as one of the most serious economic and national security challenges we face as a nation⁷. The threats come from varying sources, but the most dangerous are the sophisticated methods employed by large intelligence agencies of world-power nation-states. Admiral Michael Rogers, the Director of the National Security Agency said "I don't think there should be anybody's mind that the cyber-challenges we're talking about are not theoretical. This is something real that is impacting our nation and those of our allies and friends every day. And it is doing it in a meaningful way that is literally costing us hundreds of billions of dollars, that is leading to a reduced sense of security and that has the potential to lead to truly significant, almost catastrophic failures if we don't take action."⁸ This paper addresses the actions we can take, using advanced data management.

The Cyber Security Exercise

To help meet this threat, the authors and other conceived, designed and implemented a small unit cyber warfare system simulator.

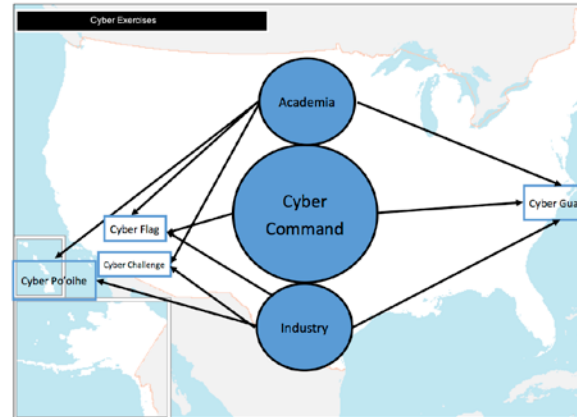


Figure 4: Cyber Exercises

They did this because of the impediments raised by doing larger exercises. It is well-known that the expenditure required to running a full-scale exercises are daunting. Millennium Challenge 2002 (MC02) was reported to have cost \$250 million dollars⁹. There have been other, smaller exercises: Cyber Flag and Cyber Guard are also reputed to have been very costly. There is also the problem of very large Internet Connection Sharing (ICS) models, organizational burdens and concomitant schedule disruptions and travel costs. These do not directly lead to enhancing the benefits of the training.

Small unit exercises have proven themselves in the battlespace simulation discipline. "America's Army" is very pervasive in today's Army and is commonly employed as a training tool. It was originally designed to be a first-person-shooter game for the US Army's Recruiting Command. Fortuitously, it was observed that trainees were playing it on-line and that play lead to enhancement of their performance and skill levels¹⁰.

In cyber warfare training, the authors assert that there is also a need for small unit an virtual environment for use in an exercise. The need for small test environments is driven by the fact that most testing is done at the small test environment level. Testing at that level will enable T&E personnel to perform frequent iterations, identify system shortcomings, and resolve inevitable mistakes in a benign environment.

The authors feel that, as the cyber war threat becomes a major focus of key decision makers, the demand for better systems, improved readiness assessment, and well tested contingency plans will become more and more vital¹¹. Because of this, several cyber exercises have been held annually. Figure 4 presents a notional view of the roles of the engaged organizations' players in these evolutions.

This requires collaborative efforts from all segments of the defense community: military, contractors and academics are essential for their success. Contributions from the all of these entities and the personnel attached to them work together: the academic community providing new technology and research, the military community incorporating this into their training and operations, and industry providing the systems and maintenance to keep everything going.

The CQRTE System

CQRTE approach displayed some very attrac-

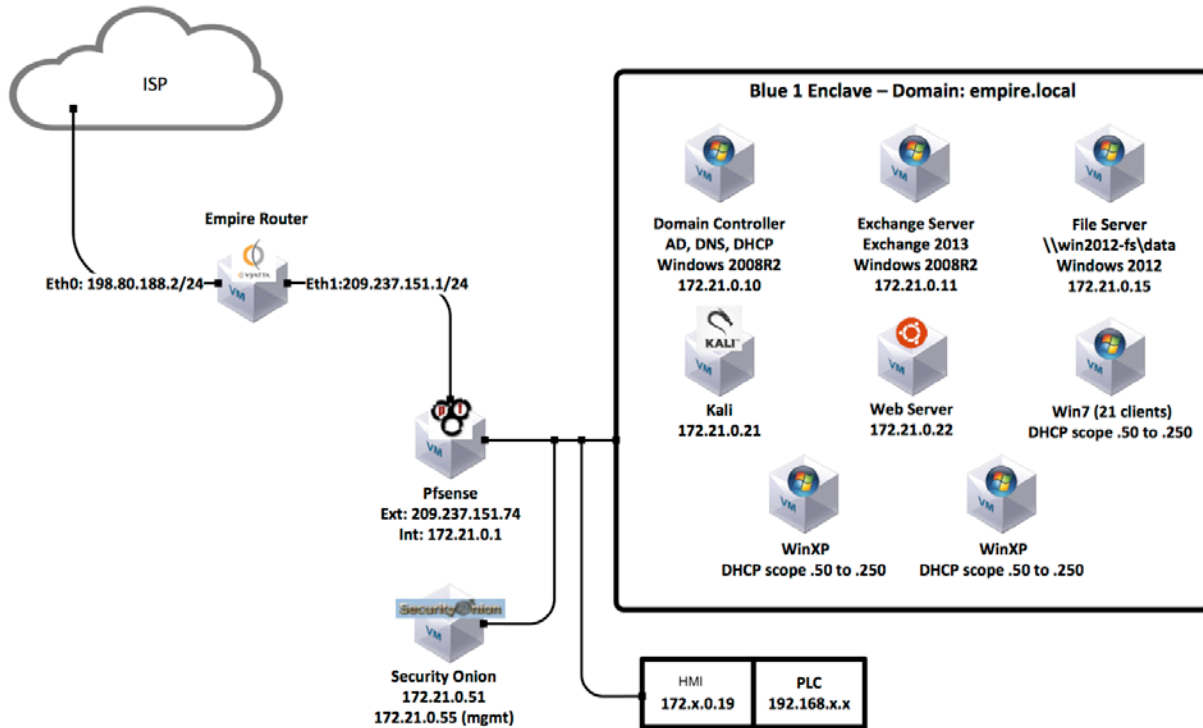


Figure 5: Notional Diagram of Cyber Challenge Network

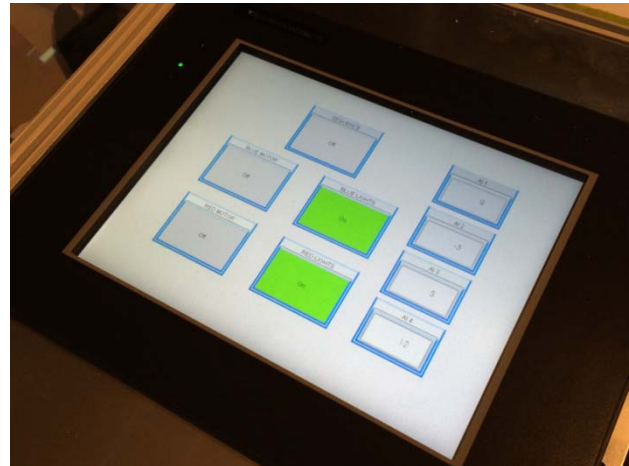
A unit of the Air National Guard led a cybersecurity training simulation team that included the University of Southern California's Information Sciences Institute (USC/ISI) and a National Laboratory. They developed the Cyber Quick-Reaction Training Environment or CQRTE (pronounced *kwerty*). CQRTE was carefully designed to be multi-disciplinary and to accept a plethora of organizations into an environment to enable small-scale non-kinetic exercises. These would effectively model warfare principles and doctrines within the context of cyberspace and cyber warfare operations. The system required very few hardware resources, as shown in Figure 5.

itive characteristics, in that it leveraged low-cost and commonly available commodity hardware and no-cost open-source software. The system automated exercise setup and teardown, as well as providing scalable exercise configurations. This system that is automation driven and it supports a diverse range of training objectives. On the data side, it assured logging and analytics for complex data and enables integration with the ICS systems. The success and wide-spread adoption of the envisioned CQRTE can serve as a functioning example for those combatant organizations whose mission-set calls for continuous training, as well as making effective use of modeling and simulation to develop the tools and tactics in the cyberspace domain. It was a replica model with which squadron and battal-

ion-level organization can design, organize, and host cyber exercises with minimal cost.



A



B

Figure 6 - (a) Dual airport runways and radar control towers, (b) PLC flyaway kit providing power to the airport runways.

The Data Management Challenges

This initiative has a very precise focus, i.e. the simulated environment has to be very effective if run by just a small unit, but must be scalable to serve a collection of units if a state-wide or region-wide exercise is planned. All of this must be done with minimal hardware and software costs, excellent security, and easily mastered technology. So, flowing from that, there are a number of challenges that faced the data-management team. They will be listed but briefly, knowing that they will be familiar to the vast majority of the readers of this paper.

The major challenge was to understand the needs of the users, all of the users and that will be addressed in the next section. Each new project will present a new set of user goals and even old projects will see user needs change over time. Careful attention must be paid to meeting this challenge and reassessing it periodically.

The second challenge was providing a flexible storage mechanism capable of storing a variety of cyber events without extensive modification or programming. Previously, in the JESPP experiments the message structures, including message type and message fields, were defined formally in Federation Object Models (FOMs) several months ahead of the actual exercise. This enabled us to use relational databases with fixed schemas. For Cyber Guard the space of cyber events need to be logged are much more fluid and difficult to bound. These events emanate from range of sources including Linux and Windows Operating System events, file system modifications, network traffic and application-level events.

A third challenge was the one of collecting the data effectively without bogging down the simulations itself. Fortunately, the experience from the JESPP project prepared the team well to address this challenge. Bandwidth is still critical and connectivity fraught with limitations and inconsistencies. The cost of disks however is modest and it is possible to store a

lot of the data on the machine generating that data, recovering it for analysis and use via the techniques developed in the JESPP Project¹².

A fourth challenge was that of using data management techniques that were easily within the grasp of non-computer science technical personnel, many of whom are excellent coders, but haven't the background in database design and use that comes with formal training in computer science. With a little attention to detail, the data system was designed to make it accessible and maintainable by the unit staffs.

A fifth challenge was quantifying the level of security required. While the work being done was not necessarily of high levels of classification, one certainly would not want potential foes gaining easy access to the exercise, where they could either assess the level of readiness of the unit, hence its vulnerabilities or intrude in malicious ways to implant malicious codes or disrupt training.

A final challenge was providing a flexible analytical framework to query the stored data discover and link relevant over time events to discover attackers, especially advanced persistent threats (ATPs) who employ slow and deliberate methodologies to avoid detection. The system should provide query facilities to discover attack surfaces used to enter the system, attempts by the attacker to escalate privileges, attacks on adjacent systems, attacker command and control, and overview of systems and resources compromised by the attacker.

Assessing the Users' Needs

For the design of the JESPP project for JFCOM, many of the authors, Yao, Davis, Tran and Lucas, sat with the analysts for many eight-hour shifts, responding to computer and communication issues, but becoming inti-

mately familiar with the operators, their issues, their skills and their needs. Then, Dr. Yao in particular, they were able to assess the best way to conceptualize, design, implement, and modify the data management system. That luxury was not afforded to the team in the Cyber Challenge system design. In this case, the internalized experience of the personnel on the team, Lt. Col. Hire, Major Tran, and Lieutenant Castello had to provide the "cyber security lore" necessary. That was then combined with generalized knowledge of cyber security held by the rest of the team.

This technique was then incorporated in the design, the testing of which be discussed below. The test provided an opportunity to both validate the choices of the designers and to provide feedback for future enhancement and modifications¹³. There is always tension in this process among the drives to provide the user with what they are familiar, to provide a more effective approach, and to program in the most elegant way. The authors have observed on many programs that serendipity also contributes in the form of operators discovering a new way to use code functionality that was neither identified nor intended by the coders and neither recognized nor requested by the user. Designers, coders and users all need to be alert for such opportunities. Any such insights must be noted, memorialized and disseminated immediately, in the form of brain-storm notes, *i.e.* free from constraints as to defensibility, grammar, spelling, or ego-involvement. The object is not to lose valuable design information and deny to the users' a capability that may have been critical.

Surveying Candidate Techniques and Technologies

Having laid down the foundations described above, a design team should follow accepted good practices in formulating, designing, im-

plementing, and testing the code. This team is oriented toward developing code in research settings and are therefore more inclined to follow accepted procedures in that environment.¹⁴ Other team may be familiar with their own discipline's standards of practice and would be similarly well advised to follow those. This team is aware as well of the insidious pressure to learn the latest trick and implement the latest solutions and therefore are assiduous in their evaluation of competing techniques based on the utility to the user, the economies presented, and the security assurance issues.

We surveyed a range of candidate technologies, each with its strength and weaknesses. Relational database technology is the most mature with well-defined SQL query language and with solid open-source implementations (such as Postgres and MySQL). However, the data storage provide by relational databases are relatively inflexible, since they require predefined data schemas. Recently, noSQL storage technologies have emerged to overcome weaknesses in relational databases. The technologies include column-oriented databases, document databases, and graph databases. Column-oriented databases are similar to relational databases, except that the relational tables are stored column-wise, instead of row-wise. Column-wise storages are better suited for data analytics, since they provide efficient column attribute sequential scans, which can be used to gather statistics (e.g., mean and standard deviation). Row-wise storages are better suited for fast atomic updates. However, both still require table schemas that need to be predefined. Document storages explicitly address the inflexibility of schemas by storing data as *documents*, where each document consists of arbitrary attribute-value pairs. However, for efficient querying the types of query has to be known in advance as to properly define the nesting structure of documents and their inter-links. In our case we

do not know the range query need to detect ATPs. Graph database provide the ultimate flexibility in relational queries by storing entities/objects as nodes in a graph and relationships/attributes as bidirectional edges connecting the nodes. The bidirectional nature of edges makes the need to predefine the nesting structure of documents unnecessary. However, scalability is a potential issue for graph databases, since they centrally maintain adjacency lists for each node. Distributed computing techniques, like document database sharding, are more difficult to apply. Moreover, in our case discovering of attackers the concept of entities/objects are more nebulous. The various activities of an attacker are not initially attributable to that attacker, or they activities may be misattributed. To support reasoning on attacker/activity attribute graph-level operation support is needed.

Implementation

We decided to use a document storage backend for our implementation by favoring flexibility and scalability in storage over query capability. Document storage provides the flexibility to store arbitrary events without the need to predefine the data schema. The document storage sharding enables us to capture and store all events even during high load. We plan to address query deficiencies in data post-processing step after the event as we better understand the attacker operating modes and procedures.

The specific document database we choose for our implementation is MongoDB. MongoDB is a widely-used open-source document-database implementation that is easy to install and to use. It stores documents in JSON format, which is an industry standard. MongoDB is agnostic to the content of the JSON documents. We choose to use Fluentd to feed event data to MongoDB. Fluentd is specialized tool for event logging. It is capable

of interfacing with a wide range of event loggers. In our case we connected NXLog and Sysmon event loggers to Fluentd. All of the Windows virtual machines in this instantiation of CQRTE were instrumented with Sysmon and NXLog. Sysmon can be downloaded from the web¹⁵ and does an excellent job of recording detailed Windows process events, including process creation, network connections and changes to file creation times. Each process is given a globally unique id (GUID) to allow identification and correlation of events when the operating system reuses process IDs. Each session is assigned a GUID as well to enable grouping of events from the same logon session. These Sysmon events are stored in the Windows Event Log.

NXLog subscribes to all the events in the Windows Event Log and can also be downloaded from the web¹⁶. It formats all of these events in the JavaScript Object Notation (JSON) format, which is an open standard, defined in both RFC 7159 and ECMA 404. It then sends these events remotely to the Fluentd which also is an open source data collector, and can be downloaded from the Fluentd web site¹⁷). In our setup there is a single Fluentd process collecting events from all Windows NXLog processes. Finally, Fluentd stores the JSON events in the persistent document store MongoDB. This software can be downloaded from the MongoDB web site¹⁸.

A Test Exercise to Validate Choices

The authors organized and managed a test of the system, including the data management sub-system in early February 2015. The test was called Cyber Challenge 2015 and was hosted by the 261st Network Warfare Squadron (NWS) in Van Nuys California. Over 50 personnel participated and they were orga-

nized into five teams. The events modeled were drafted as a potentially harmful situation that would endanger national security. The personnel were a team crew commander, who was an officer, and a battle commander. The rest of the participants were enlisted personnel from reserve units.

The exercise ran well and was considered a great training success. Vast amounts of data were efficiently collected, stored temporarily, then archived in a way that made their retrieval easy and intuitive.. These data will actually be accessed by the reservists in the future to further analyze the exercise.. All the data management mentioned above will be needed in the future. This need is predicated on the finding that, in a two-day small unit exercise, Cyber Challenger 2015 generated 1.6 terabytes of data, all of which was logged, identified and archived.

Evaluation

At the time of the submission of this paper, the evaluative process was still on-going. It is hoped that more progress will be reported at the presentation session late this Fall in Orlando. There are several thrusts to the Evaluation. The first is ascertaining if the data was collected locally without interfering with the simulation and concomitant training. The second is reviewing user needs and satisfaction with the system. Initial reactions from operators and officers alike were much more enthusiastic than was hoped and more formal assessments are planned for the future. Thirdly, there will be an effort to capture lessons learned from operators and officers, with an eye toward improving or enhancing the system. Fourth, the scalability of the system will be investigated, considering both region-wide exercises and enabling single-player exercises and training. Among the last articulated examinations will be collecting and analyz-

ing data on the ease with which the system was installed, managed and modified by the technical personnel. Finally, a tiger-team approach will be established to seek out and identify any areas of potential improvement that have been over-looked.

Suggestions for Using this Approach

Some of the more important lessons learned over the last decade and a half have distilled and presented above. The development of useful programs and their wide accessibility has at the same time provided easy access to the power of computation and induced reliance on pre-coded system approaches. The authors have shown that using well-vetted techniques can produce more optimal system that pass the test of cost/benefit analyses. Further, the use of open-source software reduces the total cost of the system, making it affordable by even small units with virtually no discretionary budget. The open source community also provides a good exemplar of using enthusiastic volunteers to do significant portions of the coding. Some of the scenario mock-ups and analytical modules for Cyber Challenge were conceived, designed and produced by reserve personnel on their own time.

One issue is of course security. There is an on-going and vociferous debate about the relative strengths of security between proprietary and open-source programs. Alan Boulanger of IBM takes the position that there are codes on both sides that are more secure than the other side of the Proprietary/Open Source divide.¹⁹ His position seems to be that one must do a de novo analysis of the security of any code irrespective of its origin. He supports the idea that the two competing areas are about evenly matched in security. Further security can be insured by “air gap” isolation

or by authorized encryption for connectivity to remote sites.

Other Potential Areas for Instantiation

The authors enthusiasm for this technique may color their seeing other opportunities for the use of these approaches in other area, but it does seem that virtually any small unit with a mission that entails the use of data, the response to outside impacts, and the need to provide an environment for training and readiness assessment would be well advised to consider these approaches. Intelligence units, cryptologic personnel, planners, combat units, communications support organizations and many others could easily adapt this paper’s suggestions to their own needs and capabilities. The availability of computer-savvy technical help is becoming so widespread that it would be hard to imagine a unit without significant computer science capabilities in their current staff. Should such integral assets be thinner than desired, other neighboring units would undoubtedly have the available expertise. It is the opinion of some of the authors that real activities like implementing a new training system produce excellent leadership and team training opportunities

Outside of the military context, these approaches could be applied to any computation and communication laden efforts in any field. Emergency preparedness, public safety, medical responses, and event planning would all fall under the aegis of the above described system.

Conclusions

Cybersecurity is vital and ensuring it is daunting. Testing of Cybersecurity systems can be costly, risky for security and disruptive, especially in very large exercises. There was a

dearth of technology available for small test environment Cybersecurity exercises. Using the expertise developed on other large-scale distributed computing projects, the team met the need of small units by producing a very effective training scenario simulator. That system and approach can now be used to provide the Test and Evaluation community with a scalable small test environment capability. It can use locally available computing power, low- or no-cost software and careful planning to meet the needs of the T&E personnel. Data management will be of paramount importance, but the approach used for the training system has established a reliable and well marked path. The small unit exercise collected around one terabyte of information daily during the conduct of the training; similar volumes are expected during test. The team has designed scalable data management systems in the past to handle orders of magnitude more data flow than this, so even large tests should pose no problems. This information can be collected, characterized, archived, and made available for retrieval and further analysis. All of this can and was accomplished within small-budget constraints. The techniques employed are almost certain to be expandable to large-scale tests and extensible into other disciplines.

ACKNOWLEDGEMENTS

The authors are grateful for the support from Lt. Brian R. Castello and LtCol. Douglas W. Hire of the Cyber Communications Squadron of the California Air National Guard. They also would acknowledge the contributions of Mr. Richard Pace, Ms. T. Katy Bragg, and Dr. Jelena Mirkovic. They are particularly grateful to SSgt Daniel Cabrera for an excellent job done on the airport runway model, Lt Selga for leading the Red Team, Major Michael “Krusty” Ehrstein for overseeing the Red force. Maj Jon Dahl for headed up the exercise logistics. The Blue Team leads were Capt

Ammie Presley and Maj Michael Cardoza. We appreciate the tremendous support from Army’s CND commander, LTC James Parsons. Finally, they would like to acknowledge the professionalism and the contributions of all of the participants of the exercise on which this paper is based. Their service was in the finest traditions of the United State Army, of the United States Air Force and of the California Air National Guard.

Ke-Thia Yao is a research scientist in the Computational Systems and Technology Division of the University of Southern California (USC) Information Sciences Institute (ISI). His primary research interest is understand large complex systems and data sets. He has conducted data management research on the JESPP project with the simulation involving millions of entities. Within the JESPP project he developed a suite of monitoring/logging/analysis tools to help users better understand the computational and behavioral properties of large-scale simulations. He received his B.S. degree in EECS from UC Berkeley, and his M.S. and Ph.D. degrees in Computer Science from Rutgers University.

John J. Tran is a Major in the California Air National Guard. He has worked at ISI, USC, the Stanford Linear Accelerator Center, Safetopia, and Intel Corporation. At USC, he focused on Object-oriented software engineering, large-scale software system design and implementation, and high performance parallel and scientific computing. Air Force tours included the White House Communications Agency and Kirkuk Regional Air Base (Iraq), where he was the Communications Squadron Commander. John is a PhD candidate in Computer Science at USC and is a programmer at the Aerospace Corporation. He received both his BS and MS Degrees in Computer Science and Engineering from the University of Notre Dame

Robert F. Lucas is a Deputy Director of the Information Sciences Institute at the University of Southern California and leads the Computational Sciences Di-

vision. He is a Research Associate Professor in the USC Department of Computer Science. At ISI he manages research in computer architectures, VLSI, compilers, and other software tools. Prior to joining ISI, he did tours as the Director of High Performance Computing Research for NERSC, the Deputy Director at DARPA's, and a researcher at the Institute for Defense Analyses. Dr. Lucas earned BS, MS, and PhD degrees in Electrical Engineering from Stanford University.

Dan Davis is a consultant for the Information Sciences Institute, University of Southern California, focusing on large-scale distributed simulations. There, he led the JESPP project for a decade. As Assistant Director of CACR at Caltech, he managed Synthetic Forces Express, introducing HPC to DoD simulations. He was the Chairman of the Coalition of Academic Supercomputing Centers and has taught at the collegiate level. Dan started writing FORTRAN pro-

grams in 1971 on Seymour Cray's CDC 6500's. He served in Vietnam as a USMC Cryptologist and retired as a Commander, U.S.N.R. He received B.A. and J.D. degrees from the University of Colorado.

Daniel P. Burns is a lifelong Systems Engineer, first with the Active Duty Navy, then SAIC, and small business. He served as Naval Chair and Professor of Practice in Systems Engineering at the Naval Postgraduate School (NPS). Captain Burns served as the as the Military Associate Dean and as acting Dean of the Graduate School of Engineering and Applied Sciences at NPS. His research interests center on analyses of both human and resource utilization in defense efforts. Captain Burns received a BS degree from the U.S. Naval Academy and an MS from the Naval Postgraduate School. He is currently finishing his dissertation for a PhD from Southern Methodist University.

¹ Gottschalk, T. D., Yao, K-T., Wagenbreth, G. & Davis, D. M., (2010), "Distributed and Interactive Simulations Operating at Large Scale for Transcontinental Experimentation", in the Proceedings of the IEEE/ACM Distributed Simulations and Real Time Applications 2010 Conference, Fairfax, Virginia

² Lucas, R., & Davis, D., (2003). Joint Experimentation on Scalable Parallel Processors, in the *Proceedings of the Interservice/Industry Simulation, Training and Education Conference*, Orlando, Florida, 2003

³ Yao, K-T., Ward, C. E. & Davis, D. M., (2011), "Data Fusion of Geographically Dispersed Information: Experience with the Scalable Data Grid", *The ITEA Journal of Test and Evaluation*, Fairfax Virginia

⁴ Gertner, A. S. & Webber B. L., (1996), A Bias towards Relevance: Recognizing Plans where Goal Minimization Fails. *AAAI/LAAI, Vol. 2* 1996: 1133-1138

⁵ Kimball, R., L. M. Reeves, M. Ross, & W. Thornwaite. (1998). *The Data Warehouse Lifecycle Toolkit*. Hoboken, New Jersey: Wiley

⁶ Yao, K.-T., & Wagenbreth, G. (2005). Simulation Data Grid: Joint Experimentation Data Management

and Analysis. In the Proceedings of the 2005 I/ITSEC, Orlando, Florida.

⁷ National Initiative, (2015), *The Comprehensive National Cybersecurity Initiative*, Retrieved from: <https://www.whitehouse.gov/issues/foreign-policy/cybersecurity/national-initiative>

⁸ Cybersecurity Threats, (2014), *Hearing of the House (Select) Intelligence Committee Subject: "Cybersecurity Threats: The Way Forward"*, November 20, 2014, Testimony of Admiral Michael Rogers

⁹ van Riper, P.K. (2004), The Immutable Nature of War, Interview transcript recorded by NOVA, the Public Broadcasting System, Retrieved from: <http://www.pbs.org/wgbh/nova/military/immutable-nature-war.html> on 10 February 215

¹⁰ Jean, G.V. (2006), Game Branches Out Into Real Combat Training, appearing in National Defense, Retrieved from: http://www.nationaldefensemagazine.org/archive/2006/February/Pages/games_brance3042.aspx

¹¹ Tran, J. J., Castello, B. R., Lucas, D. W. H. R. F., Yao, K. T., & Davis, D. M. (2015). "Employing High Performance Computing to Realize a Cyber Quick-Response Training Environment." in the *Proceedings of ModSIM World*, Norfolk, Virginia, March, 2015

¹² Yao, K-T., Tran, J. J., Lucas, & Davis, D. M. , 2015 (Pending), “Effective Data Logging and Display of Small Unit Simulations,” in the Proceedings of the Interservice/Industry Simulation, Training and Education Conference, Orlando, Florida, November, 2015

¹³ Gabbard, J. L., & Swan, J. E. (2008). Usability engineering for augmented reality: Employing user-based studies to inform design. *Visualization and Computer Graphics, IEEE Transactions on*, 14(3), 513-525.

¹⁴ Stodden, V., & Miguez, S. (2013). Best practices for computational science: Software infrastructure and environments for reproducible and extensible research. Available at SSRN 2322276. Retrieved from: http://papers.ssrn.com/sol3/Delivery.cfm/SSRN_ID_2322276_code1204238.pdf?abstractid=2322276&mirid=1

¹⁵ Microsoft Sysinternals, (2015), *Sysmon V 3.0*. Retrieved from: <https://technet.microsoft.com/en-us/sysinternals/dn798348>

¹⁶ SourceForge, (2015). *NXLog Community Edition*. Retrieved from: <http://nxlog-ce.sourceforge.net/download>

¹⁷ Fluentd, (2015). *Fluentd | Open Source Data Collector*. Retrieved from: <http://www.fluentd.org/>

¹⁸ MongoDB, (2015), *Downloads: MongoDB*. <https://www.mongodb.org/downloads>

¹⁹ Boulanger, A. (2005). Open-source versus proprietary software: Is one more reliable and secure than the other?. *IBM Systems Journal*, 44(2), 239-248.