Dynamically Correlating Network Terrain to Organizational Missions

A. E. Schulz*, B. David O'Gwynn[†], J. Kepner, and P. C. Trepagnier
MIT Lincoln Laboratory, 244 Wood St., Lexington, MA 02420

ABSTRACT

A precondition for assessing mission resilience in a cyber context is identifying which cyber assets support the mission. However, determining the asset dependencies of a mission is typically a manual process that is time consuming, labor intensive and error-prone. Automating the process of mapping between network assets and organizational missions is highly desirable but technically challenging because it is difficult to find an appropriate proxy within available cyber data for an asset's mission utilization. In this paper we discuss strategies to automate the processes of both breaking an organization into its constituent mission areas, and mapping those mission areas onto network assets, using a data-driven approach. We have implemented these strategies to mine network data at MIT Lincoln Laboratory, and provide examples. We also discuss examples of how such mission mapping tools can help an analyst to identify patterns and develop contextual insight that would otherwise have been obscure.

1 INTRODUCTION

Situational awareness of cyber assets, as well as their function in supporting the organizational missions, is crucial to mission assurance, both for prioritizing cyber key terrain (KT-C) to defend and for understanding the attack surface presented to an adversary. Equally critical is understanding an organization's sources of data, and how those data sources are utilized in support of the mission. The resources for defense of cyber assets are always limited. An organization can only allocate those resources effectively if they are aware of their cyber assets, and how those assets support the organization's goals.

Thus, mapping cyber assets onto organizational missions is critical for situational awareness, risk assessment and resource allocation. Many researchers, both in the military and in industry, have been developing manual processes to meet this critical need [2, 3, 4, 5, 7, 13, 14]. Manually produced maps are better than nothing, but they have severe disadvantages. They only consist of a single snapshot in time, and do not dynamically evolve as cyber assets are reshuffled within an organization. Also, they do not scale. As cyber assets become more multi-purposed and mobile, the timescale for reshuffling critical infrastructure will likely become so short that a manually assembled map may become obsolete even before it has been completed. Automating the process of mapping between network assets and organizational missions is highly desirable, but technically challenging because it is difficult to find an appropriate proxy for asset importance within available cyber data. However, although full automation may not be possible, most organizations have access to data sources that can be used to partially automate the mission mapping process, providing a starting point which can be manually refined.

Identification of appropriate live data sources that act as a proxy for organizational missions is an open research question. In this work we focus on the use of financial data to serve as a proxy for mission. Section §2 discusses our basic methodology. In section §2.1, we outline the technology goals for a suite of mission mapping tools developed at MIT Lincoln Laboratory. For brevity, we will refer collectively to this suite of utilities as PCAMM, or Person Centric Automated Mission Mapping. In section §2.2 we discuss one possible implementation. In section §2.3 we outline some of the applications of PCAMM, with some examples drawn from data at MIT Lincoln Laboratory. In section §3 we draw conclusions, discuss the drawbacks and limitations of this approach, and outline open research questions for the future.

2 METHODOLOGY

The key challenge in this work is to develop mapping technology that is automated and data driven. There are three principal insights that guide our basic approach. The **first** and most fundamental is that the link between network assets and organizational missions is the workforce: people are required to carry out tasks and accomplish goals, and people are also the users of an organization's network. If the people can be mapped to the organizational missions, we can pivot through them to the network assets that they use in executing their tasks.

The **second** insight is that finance data provides a mechanism to find out what the people in an organization are doing without manually interviewing them. All government and business entities have a charging structure. The money they spend is subdivided according to what the money is for i.e. the missions and programs. The money is also distributed to people who work on those missions and programs. Therefore the finance data provides a link between the people and the missions.

The **third** insight is that different questions will require the mission map to be enriched with different data sources. The finance data providing the link between people and missions is not "big" in the sense of big-data. These can exist entirely in memory, even for a very large organization. However, enriching the mission map with all the network log data at once cannot be approached with the fast in-memory hash-map technique used for the unenriched mission map. Since we need a forensic tool that is lightweight and fast, we engineer a way to leverage the big-data assets and pivot on the fly to whichever network data sources help answer the specific question.

2.1 Technology Goals

The development of our suite of mission mapping tools, which we call PCAMM, is guided by a number of overarching technology goals. The intent is to map network and cyber infrastructure to organizational missions. The tools should provide a method to identify mission critical assets: key users and accounts, as well as key infrastructure. They should provide insight on the mission impact of compromised accounts, compromised or unavailable machines, and both the network and organizational role of Information Technology (IT) systems of interest.

Although the PCAMM software is not limited to visualization of the data, it is designed to conform to Ben Schneiderman's mantra [11]. It should provide overview data and big-picture mission context; it should provide search and filter capability; and it should provide details on demand, allowing the user to specify new layers of enrichment on the fly. The mission mapping tools are designed to allow real-time mission-driven sense-making and forensic capacity.

*Corresponding author. e-mail: alexia.schulz@ll.mit.edu

[†] Present address: Belhaven University, Jackson, MS 39202

Finally, PCAMM should be as automated as possible. The goal is to use data to identify key network infrastructure, so that the tools can provide real-time context that doesn't get stale as a network evolves.

2.2 Implementation

2.2.1 Prototype

We have built a prototype implementation of the automated mission mapping tool. We emphasize that this is just one possible implementation; if an organization has the requisite data available in their environment, these ideas could be executed in any of a number of ways. Our implementation consists of three distinct pieces: a data layer consisting of an Accumulo database; a knowledge engineering layer consisting of a domain specific language (DSL), implemented in Python, used to interface with the database, and a mission mapping interface, written in Python, that leverages the DSL to build the mission map and enrich as needed. We focus on the third layer in this work, but again wish to emphasize that the Python-based approach utilized in our proof-of-concept implementation was chosen tactically for speed of development, as it interfaced easily with LRNOC's existing LLCySA platform [10]. Many other frameworks (e.g. the ELK stack or a SQL database) are also possible.

The primary class in the mission map utility is a person. A person has unique attributes, such as a user id, an email handle or a network account name. A person also has attributes that are shared with other people, such as a research group or a job title. Some of the attributes are one-to-one, such as person's first name. Others are one-to-many, such as the programs a person works on. In this latter case, the value stored in the attribute is a list. These person objects are combined to form another type of class: a person-list. The person-list is fundamentally a hash-table which is keyed on one the the unique attributes of the people in the organization, such as a badge id number. Most of the operations carried out by PCAMM are actuated by manipulating person-list objects. The key feature of the person-list class is the enrich method, which takes any dictionary of data keyed on people and uses it to add attributes to the people in the map.

As with any data store, one critical tool built into PCAMM is the slicing tool. The slice feature allows the user to filter the mission map on any attribute of a person, and return a slice whose members either all have, or don't have, that attribute. It is useful to break the mission map of an organization into sub-maps (slices), each mapping out some subset of whole enterprise. The sub-maps can be unique, or they can be overlapping. Making a slice for every possible value of an attribute is a process that we call organizational breakdown. A breakdown is stored as a dictionary structure where the keys are the N_{val} possible values the attribute can take, and the value is a sub-map: i.e. the corresponding person-list. The breakdown computation is an embarrassingly parallel process, and the computation time scales approximately as

$$t_{\text{breakdown}} \sim O(t_{\text{sg}} * N_{\text{val}}/N_{\text{proc}}) \tag{1}$$

where N_{proc} is the number of processors the user has available, and t_{sg} is the computation time to generate one slice, which scales approximately as the number of people N_p populating the map: $t_{\text{sg}} \sim O(N_p)$.

2.2.2 Data sources

The mission map is populated with data from the organization. The data sources are divided into two categories, organizational data sources and network data sources. The organizational data sources are used to build the un-enriched mission map, which we describe shortly. The network data sources are used to enrich the mission map, at the operator's discretion, with log data that is typically available in a Network Operations Center (NOC). We emphasize, however, that any data associated with people can be used to enrich the map. The data used in the prototype implementation PCAMM is data assembled from the MIT Lincoln Laboratory network, which is stored and accessed in the Lincoln Research Network Operations Center (LRNOC) [6, 9]. The data is accessed via the Lincoln Laboratory Cyber Situational Awareness (LLCySA) platform [10].

One of the organizational data sources is phone book or directory data; usually Lightweight Directory Access Protocol (LDAP) data in LDIF format (LDAP Data Interchange Format), though the directory data could be in any format. The directory data typically provides names, identification numbers, usernames, email handles, phone numbers, office locations, job titles, as well as information from the organizational chart such as the divisions and groups for the people in an organization. The directory data is used to create a person-list that will form the backbone of the un-enriched mission map. We have enhanced the map by binning the job titles into broader roles within an organization. For Lincoln Laboratory, the roles we use are research, technical support, administrative support, leadership, information technology, security and students. This assignment of titles to role categories is one example of a manual processes that fits into the mapping framework. For our work, we have also used data from Human Resources to enrich the map with university degrees, as a proxy for subject-matter expertise. (However, in military contexts much more precise occupational specialty data are available.)

To create the un-enriched mission map, the directory data backbone is fleshed out with organizational financial data. There are two main inputs that are required for this approach. The first is the labor charging structure in the organization; for each person there is a record of which programs were charged in each month, as well as the fraction of time that was charged to the program. The second is finance data on the mission allocation of money used to fund each program, that is, which mission area is supported by each program. Although it is possible that this mapping may need to be manually assembled, it is common practice in most business offices to maintain such a list and often the data already exists, as it did in the case of Lincoln Laboratory. Once these data sources are incorporated into the map, each person will have a list of programs they contribute to, and a list of mission areas (and potentially submissions) that they support. This is the un-enriched mission map.

The mission mapping tool functions by taking this base-level mission map and enriching it in an ad-hoc fashion with network security data that is housed in some underlying database. In our case we have used the Lincoln Laboratory LLCySA platform [10], but as previously mentioned any queriable database structure is an option. There are several network data sources that we have incorporated into PCAMM. One of the most important are the authentication logs. These provide a mapping between users and machines that they utilize, either by directly logging on with a username and password, or authenticating with kerberos credentials. Another important source is the property tracking data, which help us to map between users and the machines they own. We have also incorporated Nessus [12] vulnerability reports, which tell us which network assets are potentially targets. Mail exchange metadata can reveal more informal communities within the organization. Other possible data sources include web proxy logs, VPN logs, IDS alerts and host-based security system data such as logs from McAffee [1]. For any of these network indicators, we can pivot through the people associated with the machines of interest to the missions and programs that are potentially affected.

2.3 Use cases

2.3.1 Overview Utilities

There are a number of utilities modeled on database queries that build on the mission map infrastructure to provide added function-

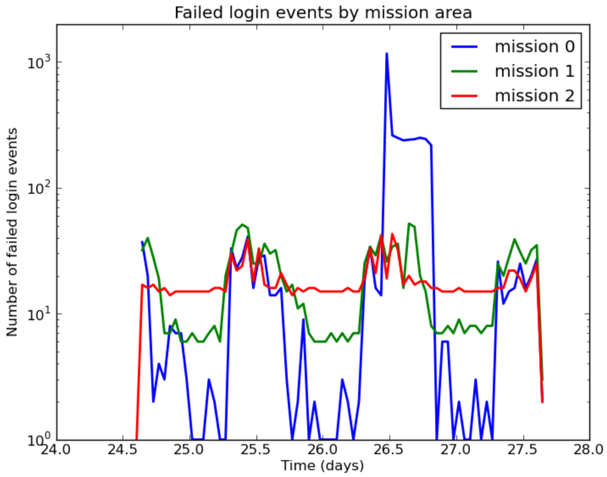

Figure 1: *Failed login event data, binned every hour, for a ∼ 3 day period in February 2014. Three of MIT Lincoln Laboratory's mission areas are represented. On Feb 26th, there was a spike in failed login attempts, and these predominantly affected one of the mission areas while leaving the other two at levels close to the historic baseline.*

ality to PCAMM. One such utility is the *getall(attribute)* method. This method aggregates all values of a particular person-attribute for people in the map, and returns a list of these values. For example, an analyst could create a sub-map whose members were owners of vulnerable machines identified in a Nessus scan. The *getall(emails)* method could be used to retrieve a list of the owners' email handles, quickly generating a distribution list to whom patch information should be sent.

A related utility is the *gethist(attribute)* method. This method returns a dictionary: the keys are each possible value of an attribute and the values are the number of people who possess that value of the attribute. Calling *gethist(missions)* will enumerate how many people in an organization work on each mission area. Calling *gethist(loginfails)* on each sub-map in a mission breakdown can indicate whether any particular mission area is the target of a password spraying attack. For convenience the the *plothist(attribute)* method displays a plot of the histogram.

The PCAMM software allows the user to quickly enrich external data sources with programmatic or mission context. Figure 1 demonstrates this: the raw data are active directory log events of failed login attempts at MIT Lincoln Laboratory between 24-27 Feb 2014, which have been enriched with mission context using PCAMM. We only plot three of the mission areas in Figure 1. On Feb 26^{th} around noon, there was an event generating failed login counts several orders of magnitude higher than the baseline. Figure 1 shows that these events potentially have more impact on one of the mission areas than the others. This is only one example. Since PCAMM allows an analyst to enrich any new data source with mission context, any event data associated with hosts, ip addresses, or people can be enriched with programmatic or mission context in this way.

The data in the mission map is inherently graph data, with nodes and edges that connect the various attributes such as ip addresses, missions and people. However graph representation of the data often does not convey sufficient contextual information to be useful. Our approach to this difficulty is to use a configurable treemap to convey context in a flexible way. The breakdown method described in section 2.2.1 can be applied recursively. The result is the creation of a tree data structure that provides an overview of the components that make up an organization. The *tree([list of attributes])* method provides a convenient way to explore the data to understand large trends. An example, shown in Figure 2 with data from MIT Lincoln Laboratory, might be to break down the organization by mission and then by ip, to see which assets are most commonly utilized by each mission area. It is particularly powerful that the user can specify whatever branching order best answers the specific question.

Figure 2 reveals that some assets are commonly used by everyone in the Laboratory, whereas others are used only by specific mission areas. The tree utility accepts a list of values to ignore, in case the user would like to exclude the mail server or other common assets. The method will build trees to arbitrary depth, which is combined with a zoomable functionality in the visualization, that allows the user to descend down to successive layers. Only two layers of the tree are simultaneously visible in a given view. For example, if we had built this tree with branding order *[missions, ips, nessus-plugins]*, the nessus id numbers would be hidden in the top level view. However the analyst can click in any of the mission areas. The visualization would zoom in and show one ip per colored area, with subdivisions depicting nessus plugin id.

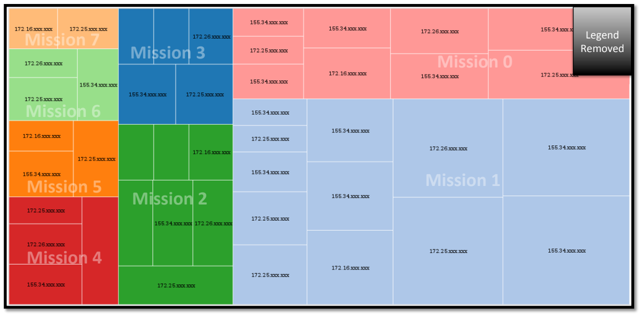

Figure 2: *The treemap functionality allows an analyst to get overview context of the data in the mission map. This example was generated using data at MIT Lincoln Laboratory to build a tree with branching order (missions, ips). Tree data structures can be built to arbitrary depth, but the visualization shows only two layers at a time. Deeper layers are revealed by clicking in a colored area. The legend and actual mission names have been obfuscated.*

One of the goals of PCAMM is the discovery of relationships between network entities, people, missions and programs. To this end there are two useful utilities that quantify the extent to which entities are "connected," the correlation method and the pattern method (discussed in section 2.3.2). The correlation utility is best explained in terms of a Venn diagram. If two populations overlap, one can compute the conditional probability that a person in set A is also a member of set B with Bayes' theorem (e.g. [8]).

$$P(B|A) = \frac{P(A \cap B)}{P(A)} = \frac{N_o/N_T}{N_A/N_T} = \frac{N_o}{N_A} \tag{2}$$

Here N_o is the number of people in the overlap region of the Venn diagram, N_T are all the people in the organization, and N_A and N_B are the number of people in groups A and B, respectively. This probability quantifies the correlation of the two properties represented by A and B within the organization, and is equal to 1 if the two properties correlate perfectly (i.e. complete overlap of the Venn Diagram). The method *correlate(attribute1,attribute2)* will compute the correlation of every value of one attribute with ever value of another. Computing the autocorrelation will quantify the clustering within an attribute, for example, correlating programs to programs will quantify what fraction of personnel are shared between any two programs.

Because the correlate function is many-to-many, it doesn't lend itself well to visualization. Therefore, a *plot-*

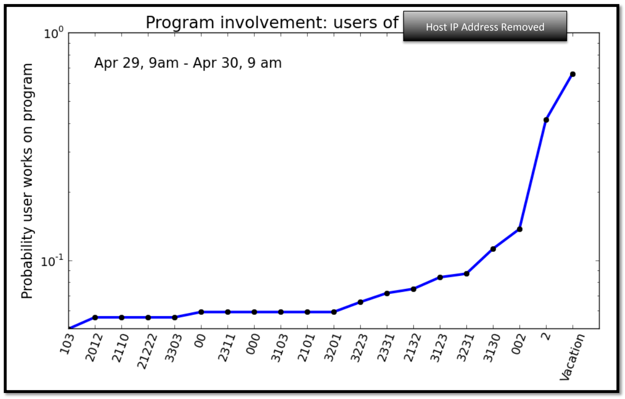

Figure 3: *The probability that a user of the host at ip xxx.xxx.xxx.xxx charges to a particular program is shown for users who authenticated between April 29th and 30th. This is real data from MIT Lincoln Laboratory, although the ip and the program numbers have been obfuscated.*

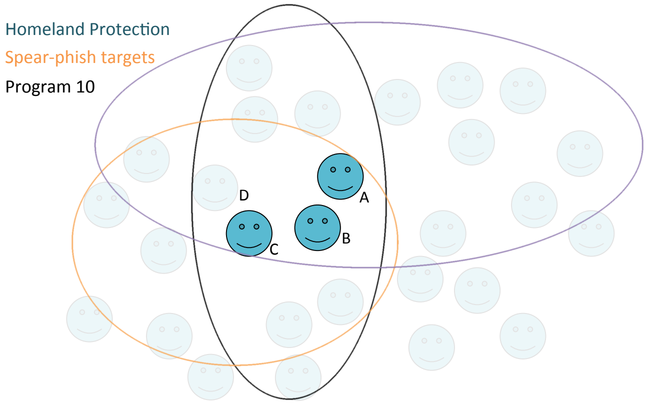

Figure 4: *If A, B and C are associated with IT systems of interest, the pattern discovery methods in PCAMM allow an analyst to determine what the people have in common, and the pattern matching method facilitates the discovery of person D, whose assets may also be compromised.*

Attribute	Value	# People / 7	Unique
missions	Space Control	22	No
sponsors	Other DoD	14	No
gender	Female	7	Yes
role	Research	7	Yes
title	Technical Staff	4	Yes
propertylocations	Z1-123	4	No

Table 1: *Output of the pattern() method on a sub-map of MIT Lincoln Laboratory data. The sub-map corresponds to owners and users of assets at several ip addresses. The table quantifies what the individuals in the map have most in common.*

corr(attibute1,value,attribute2) method will display the correlation of one specific value of attribute *A* with all possible values of attribute *B*. Figure 3 shows data collected in a 24 hour period on Apr 29th 2014. The plot shows the probability that a user of the host at the obscured ip address works on a given program along the $x-$axis. The data depicted are real, but the ip and program numbers have been obfuscated for MIT Lincoln Laboratory operational security. This type of analysis could be useful in quantifying which programs are potentially at risk, in the event the host at this ip address were compromised.

2.3.2 Pattern Utilities

A critical part of situational awareness is the ability to recognize and match patterns in the data. For example, if network assets are found to be compromised, it is reasonable to inquire whether a particular individual is associated with them. It is also useful to know whether these assets work in concert to support some programmatic or mission function. Such patterns may help an analyst to decipher the root cause of the infection, and whether there is a particular target of the compromise within the organization. Pattern matching is critical for searching out other individuals who may be affected, or identifying other network assets potentially at risk.

The cartoon in Figure 4 illustrates this idea. Suppose malware is detected on the laptops of persons A, B and C. The pattern method can help identify what these people have in common. In this example, they all support the Homeland Protection mission area, they all charge to a particular program, and they have all been targets in a recently identified spear-phishing campaign. Discovering this information is useful for potentially tracing the source of the infection to a phishing attack, identifying which program and mission areas are potentially impacted, and locating person D. This person has many features in common with A, B and C; their assets may also be infected, or person D may inadvertently have been the vector propagating the infection.

The *pattern()* method can be called on the mission map or any sub-map. It systematically examines data with which the map has been enriched. It returns a sorted list of tuples, containing the attribute, the most common value found in the sub-map, and how many people share that value. An example is shown in Table 1. The *pattern()* method was used on a sub-map created for this example, which corresponds to owners and users of assets at several ip addresses at MIT Lincoln Laboratory. There are seven individuals in the sub-map, and these results show that collectively they work on 22 programs in the Space Control mission area, many of which are sponsored by Other DoD sponsors. Owners and users of these machines happen all to be female members of the research staff. If these machines constituted a list of compromised assets, this pattern might help an analyst determine that the vector for the threat was somehow connected to a professional association of technical women. The analyst can use the map technology to explore this hunch further. For example, she may be curious to see if these women have recently attended some conference in common. She could use the *enrich()* method to add travel data to this sub-map, and potentially discover how many of these individuals had attended a recent conference on women in STEM fields.

The functional inverse operation of the *pattern()* method is the *pattern_match()* method. A user supplies a template with feature values, and the map is scanned to return a sub-map whose members match the template. An example template used identify person D (as well as A, B and C) in Figure 4 might be (missions = Homeland protection, programs = 10, speardates != None). Pattern matching is accomplished by repeatedly applying the slicing algorithm. It is an example of a utility that would be easier to optimize if the mission map were implemented directly in a database, rather than the in-memory prototype discussed in this work.

3 Discussion

We present in this paper an important first step in dynamically correlating cyber terrain to organizational missions. Such a mapping is critical for enhanced network situational awareness, for developing forensic insight, for assessing risk and optimally allocating resources to defend mission-critical assets. Our approach of incorporating finance data and identity stores with passive network monitoring and log data demonstrates that an approximate map can be built dynamically, providing insight that would not otherwise be available. However, there are limitations to this approach. First, this implementation uses authentication data as a proxy for asset importance. While frequency of use is one important indicator, it will overemphasize the importance of some assets, and will potentially miss other critical assets such as servers and routing infrastructure, whose use does not typically require authentication. One important extension of this work that will greatly mitigate this inaccuracy is the addition of logs from routers or switches, that lend insight into internal connections between machines. With this new source of information each person could be associated with a list of hosts and/or ip addresses that they directly use, as well as a second layer of hosts/ips observed to connect with the first layer in netflow data. Not only will this provide more complete insight into the assets used in a given mission area, it will also be critical for baselining internal network connectivity, deviations from which can help indicate potential insider threats or lateral movement.

Another drawback of this approach is the heavy reliance on finance data to identify the missions and programs. While this mapping is likely to exist for most organizations and business entities, it will be far more accurate in some cases than in others. Following the money will always provide an approximate proxy for organizational mission structure, but it would be best to combine this with other indicators of a person's role or work function. These indicia will in many cases be directly available, but may be augmented by e.g. semantic analysis of their documents or email.

Another potential improvement to the implementation presented here will be to add data dependencies of the various missions and programs to the map. This will provide a secondary indication of asset importance, particularly for database and server infrastructure. It will also help interpret the risk associated with adverse events, such as a discovery of data exfiltratration, to the missions and programs. We reserve an analysis of mapping data dependencies for future work.

Despite these imperfections, the PCAMM implementation has demonstrated that pivoting through employees to filter for attributes of interest is a highly effective way to increase network situational awareness and convey mission context. The PCAMM software provides

- Overview
  - Display histograms for a program, division or mission
  - Create treemaps showing the distribution of subgroups and sub-subgroups
  - Compute statistics and distributions for aggregated quantities
  - Identify rare attribute values
- Zoom and Filter
  - Filter on a particular feature to create a sub-map
  - Correlate between any attribute and any other
  - Identify individuals who meet some profile
- Pattern identification
  - Identify commonalities between persons (or machines) of interest
  - Identify others assets that fit the pattern
- Details on Demand
  - Retrieve data for any person or group of interest

Mapping network assets onto organizational missions is vital for identifying infrastructure and quantifying its mission impact, identifying interdependency of mission components, identifying patterns and finding other assets that meet a profile. Mapping informs the optimization of resource allocation, helps to quantify risk, and provides the basis for mission assurance. We have demonstrated some initial capabilities toward the ultimate realization of dynamically achieving these goals, and are searching for appropriate use-cases on which to test and calibrate the method.

Acknowledgements

The authors wish to thank the members of the LLCySA team for their support of this work. They also acknowledge useful conversations with Jeffrey Gottschalk, Bill Streilein, Bill Campbell and James Riordan. This work is sponsored by the Assistant Secretary of Defense for Research & Engineering under Air Force Contract #FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

References

[1] McAfee: an intel company. http://www.mcafee.com/us/, 2014.

[2] A. D'Amico, L. Buchanan, J. Goodall, and P. Walczak. Mission impact of cyber events: Scenarios and ontology to express the relationships between cyber assets, missions, and users. Technical report, DTIC Document, 2009.

[3] J. Goodall, A. D'Amico, and J. Kopylec. Camus: Automatically mapping cyber assets to missions and users. In *Military Communications Conference, 2009. MILCOM 2009. IEEE*, pages 1–7, Oct 2009.

[4] S. Musman, M. Tanner, A. Temin, E. Elsaesser, and L. Loren. A systems engineering approach for crown jewels estimation and mission assurance decision making. In *Computational Intelligence in Cyber Security (CICS), 2011 IEEE Symposium on*, pages 210–216, April 2011.

[5] A. Natarajan, P. Ning, Y. Liu, S. Jajodia, and S. Hutchinson. Nsdminer: Automated discovery of network service dependencies. Orlando, FL, March 2012. IEEE, IEEE International Conference on Computer Communications.

[6] J. O'Connell. Lincoln research network operations center. MIT Lincoln Laboratory, Cyber Netcentric Workshop, June 2014.

[7] E. Peterson. Making sense of cyberspace: visualizations for analysts and decision-makers. Johns Hopkins University Applied Physics Laboratory, Visualization and analytics for cyber situational awareness, August 2013.

[8] W. H. Press. *Numerical recipes in C++: the art of scientific computing. Example book in C++*. Cambridge University Press, 2002.

[9] S. M. Sawyer, B. David O'Gwynn, A. Tran, and T. Yu. Understanding query performance in accumulo. In *High Performance Extreme Computing Conference (HPEC), 2013 IEEE*, pages 1–6. IEEE, 2013.

[10] S. M. Sawyer, T. H. Yu, M. L. Hubbell, and B. D. O'Gwynn. LLCySA: Making sense of cyberspace. *Lincoln Laboratory Journal*, 20(2), 2014.

[11] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE, 1996.

[12] Tenable Network Security . Nessus open source vulnerability scanner project, 2005.

[13] P. Verga. Defense critical infrastructure actions needed to improve the identification. *Department of Defense, Manual*, 1(3020.45), October 2008.

[14] J. Watters, S. Morrissey, D. Bodeau, and S. C. Powers. The risk-to-mission assessment process (riskmap): a sensitivity analysis and an extension to treat confidentiality issues. *The Institute for Information Infrastructure Protection*, 2009.