National Cybersecurity Institute Journal

Volume 1, No.1

National Cybersecurity Institute Journal

Volume 1, No. 1

ISSN 2333-7184

NATIONAL CYBERSECURITY INSTITUTE JOURNAL

The National Cybersecurity Institute at Excelsior College is a research center based in Washington, DC, dedicated to increasing knowledge of the cybersecurity discipline and its workforce demands. Published three times a year, the peer-reviewed *National Cybersecurity Institute Journal* covers topics that appeal to a broad readership within the cybersecurity discipline, with a particular focus on education, training, and workforce development. The manuscripts submitted to the journal are reviewed for their contribution to the advancement of applied research in the area of cybersecurity.

Submission guidelines for authors can be found at www.nationalcybersecurityinstitute.org/journal/.

FROM THE EDITOR

Welcome to the premiere edition of the *National Cybersecurity Institute Journal.* The mission of the National Cybersecurity Institute is to increase awareness and knowledge of the cybersecurity discipline and assist the government, industry, military, and academic sectors to better understand and meet the challenges in cybersecurity policy, technology, and education. To that end, the *National Cybersecurity Institute Journal* will present relevant and noteworthy articles that will serve to enlighten those with a vested interest in the cybersecurity field. In this first edition, you will find six articles from notable authors with a variety of perspectives in the field.

Shana Kayne Beach presents an article on human factors in cybersecurity, which is often overlooked because cybersecurity education and training programs center primarily on technical and/or policy curricula. Jim Jones and Carl Beisel suggest the detection of novel cyber attacks in real time is difficult due to the large volume of data available, and an uncertain relationship between raw network data and novel attacks. They present us with an approach and experimental results addressing these challenges. Alireza Aghamohammadi and Ali Eydgahi propose a new method to prevent unwanted Web robots from accessing websites. Their method utilizes five identifiers — passkey, time, Internet Protocol address lookup, user agent, and number of visits for evaluation process — of granting access to Web robots. Brian M. Mazanec writes that the global community is increasingly dependent on cyberspace, but there are no clearly agreed-upon norms for acceptable state behavior in cyberspace. He presents a paper that offers a framework to help explain how norms for cyber warfare are likely to develop.

Gordon Romney, Miles Romney, Bhaskar Sinha, Pradip P. Dey, and Mohammad N. Amin discuss the power of 'Rails,' which was selected for CSIA, at the suggestion of an industry collaborator, because it enforces good coding habits, encourages better security practices, is used in cyber tool creation, and its framework facilitates agile development and course delivery. Finally, Aftab Ahmad and Ping Wang present an analysis of security assessment of wireless LANs (WLANs) in a classified environment. The analysis is based on a technique derived from ITU Recommendation X.805.

A publication is never the work of one individual, but rather a collaboration of dedicated people who work tirelessly to produce a quality product. A great many thanks go to all the contributors, administration, and staff for their efforts to bring the *National Cybersecurity Institute Journal* to fruition. I am sure you will find this journal informative as the cybersecurity field continues to evolve. I look forward to your comments, suggestions, and future submissions to our journal.

Dr. Jane A. LeClair
Editor in Chief

Usable Cybersecurity: Human Factors in Cybersecurity Education Curricula

Shana Kayne Beach

ABSTRACT

Current cybersecurity education and training programs center primarily on technical and/or policy curricula. Although these topics contain the core knowledge set needed to develop cybersecurity professionals, graduates may not be prepared to translate their newly acquired expertise to a non-technically inclined audience or understand the risks inherent human nature. The greatest barriers to eliminating the unintentional insider threat include obtuse documentation, overly complicated network security policies, unintuitive security software and applications, and organizational culture weaknesses. This study surveyed 129 academic institutions that offer cybersecurity programs for their human factors requirements. Of the resulting data, only 2% of programs required human factors courses for graduation, 36% offered human factors courses within the department or as electives, and 62% did not offer human factors courses at all. To reduce this imbalance and reflect the high priority of human factors in cybersecurity, academic institutions should develop human factors, usability, and communication curricula to assist graduates in making security intuitive and reflexive for those outside the discipline. The article concludes by offering recommendations for cybersecurity human factors curricula development.

Keywords: academics, cybersecurity, education, ergonomics, human factors, human-computer interaction, information assurance, network security, usability

HUMAN FACTORS AND CYBERSECURITY

The *human factors* discipline, often used interchangeably with the term *ergonomics*, refers to "the scientific discipline concerned with the understanding of interactions among humans and other elements of a system, and the profession that applies theory, principles, data and methods to design in order to optimize human well-being and overall system performance" ("Definition and Domains of Ergonomics," 2013). In other words, a successful application of human factors principles and methods will increase the usability of a product or system. The *cybersecurity* discipline focuses on the ability to protect assets from cyber threats which can include the use of information assurance methods to ensure availability, integrity, authentication, confidentiality, and non-repudiation (Kissel, 2013). The application of human factors principles to cybersecurity is critical due to the human interaction that takes place constantly with any system requiring availability, integrity, authentication, confidentiality, and non-repudiation. Gonzalez and Sawicka (2002) note that 80-90% of security problems are due to human factors-related vulnerabilities. This is further substantiated by Verizon's report indicating that error and misuse are responsible for 68% of security incidents, and 29% of cybersecurity breaches are accomplished through use of social tactics alone, and ("2013 Data Breach Investigations Report," 2013). Therefore, it should be a cybersecurity practitioner's greatest priority to ensure that human factors are taken into account when ensuring information assurance needs are met. There are several aspects of human factors that should be considered when developing a cybersecurity program.

Usability

Cybersecurity programs often incorporate hardware or software solutions with which the user must interact on a regular basis. If ease of human-computer interaction is not taken into account, the user may unintentionally switch to a less secure system or disable elements of a security system. The best technology may be defeated by its inaccessibility to the user (Kraemer, Carayon, & Clem, 2009; Theofanos & Pfleeger, 2011).

Cognition and Pyschology

The process of human decision-making is extremely important to human factors in cybersecurity because it can help identify vulnerabilities to social engineering attacks such as phishing (Bowen, Devarajan, & Stolfo, 2011). Even a knowledgeable and skilled user can be persuaded into poor security decisions through natural biases and decision-making errors (West, 2008). It's important for cybersecurity practitioners to understand the cost-benefit factors and risk perception that come into account when a user is unknowingly faced with a cybersecurity challenge (Gonzalez & Sawicka, 2002).

Social and Cultural Influences

User actions are strongly affected by outside influences. If an organization prioritizes efficiency of workload completion over dedication to security requirements, members will respond accordingly by deprioritizing security (Cranor, 2007; Kraemer et al., 2009). Users must trust and understand the security policies being communicated through the organization, and the culture of the organization must make security threats tangible to its members (Nurse, Creese, Goldsmith, & Lamberts, 2011b). The study of social, cultural, and organizational influences on cybersecurity can ensure that cybersecurity practitioners are aware of the effect of communication and interaction on the systems in place.

Research Methods and User Testing

Cybersecurity systems are often designed by engineers or developers. While the program may seem intuitive to their particular thought processes, the end-user may interpret an interface completely differently. By the same token, organizational leadership enacting a new cybersecurity policy may believe the direction to be clear, but those responsible for implementing it may derive an alternate meaning. In order to prevent either scenario, cybersecurity practitioners should be trained to conduct research and user testing to ensure programs are understood and used as intended. This may include formation of multi-disciplinary design teams, active involvement of users in the development process, and iterative development cycles based upon results of user testing (Maguire, 2001; Nurse, Creese, Goldsmith, & Lamberts, 2011a). Additionally, cybersecurity professionals should be aware of the aspects of human nature that make themselves and their end-users vulnerable to exploitation and attacks (West, 2008).

These aspects of human factors are all interrelated and equally critical to the development of a strong cybersecurity system. Cybersecurity practitioners must be trained to incorporate these standards, and any reputable cybersecurity training or education program should include usability, cognition and psychology, social and cultural influences, and research methods and user testing.

SURVEY OF HUMAN FACTORS INCLUSION IN CURRENT CYBERSECURITY CURRICULA

To establish the current status of human factors education in current cybersecurity curricula, this research surveys the courses of study for various cybersecurity programs within the United States.

Sample/Population

The sample used for this study consisted of 129 academic institutions identified as "National Centers of Academic Excellence in Information Assurance Education" by the National Security Agency and the Department of Homeland Security due to a focus on "promoting higher education and research in [Information Assurance] and producing a growing number of professionals with [Information Assurance] expertise in various disciplines" ("National Centers of Academic Excellence," 2009). These institutions were selected for this study due to their level of national recognition and information assurance or cybersecurity emphasis.

Methods and Procedures

Information about each academic institution was obtained using the URL provided by the National Security Agency's list of National Centers of Academic Excellence ("Centers of Academic Excellence Institutions," 2009). The relevant master's program (usually in Information Assurance, Information Technology, Computer Science with an emphasis in security, or Cybersecurity) was selected and measured for number of credit hours required, number of human factors credit hours required, and number of human factors credit hours available as electives. When an institution did not have a relevant master's program listed, the next highest equivalent was selected. When this meant selecting a bachelor's or associate degree program, the general academic (non-cybersecurity related) requirements were subtracted from the total credit requirement.

When measuring the total course hours required for completion and multiple options were available (such as thesis or coursework-only), the lowest minimum was selected for the metric.

When measuring the number of human factors courses available as electives, only courses within the same department as the program were counted unless the program specifically identified inter-department courses were available. This means that some human factors classes offered by design or psychology departments were not included in the metrics.

For all metrics, in order to qualify as a human factors course, the title was required to include "human," "ergonomics," "usability," or "interaction" with a course description that indicated a clear focus on human factors.

Study Weaknesses

This study focused almost entirely on graduate-level programs, which tend to be very specific in nature. A future study could measure the inclusion of human factors in undergraduate-level curricula, which may cover broader topics and have increased likelihood to allow for human factors electives.

These programs were not evaluated for program quality or the content of the courses themselves, other than their recognition as a Center of Excellence. Future exploration into the quality of human factors curricula could include more qualitative analysis of the content of the courses and their application to cybersecurity concerns. The elements of usability, cognition and psychology, social and cultural influences, and research methods and user testing should be covered in any cybersecurity and human factors course.

Additionally, this study measured only entire courses dedicated to human factors materials. A future study could measure what how of all course content (including non-human factors courses) includes human factors as a priority.

Finally, this study relied entirely upon data publically available online. Of the 129 institutions surveyed, eight did not have sufficient information available on their websites to develop metrics ("George Washington University," 2013; "Idaho State National Information Assurance Training and Education Center," 2013; "Rochester Institute of Technology Center for the Advancement of Research and Education," 2013; "Southern Illinois University School of Information Systems and Applied Technologies," 2013; "United States Air Force Academy Department of Computer Science," 2013; "United States Naval Academy," 2013; "University of Detroit Mercy Graduate and Professional Studies," 2013; "West Point Cyber Research Center," 2013). An alternate version of this study could alleviate this problem by contacting schools directly and asking for assistance in completing the metrics.

Results

Table 1, presented at the end of this article, lists the complete results of the survey, including the academic institution, the total number of credit hours required to complete the cybersecurity program, the number of human factors credit hours required, and the number of human factors credit hours available.

Major Findings

Programs without Human Factors Curricula. Of the 121 academic institutions with complete data available via a public website, 75 (or 62%) neither required human factors coursework nor offered electives in human factors. These programs averaged 32.6 credit hours required for completion. This data means that the majority of cybersecurity graduate programs do not require or offer any courses specifically dedicated to human factors, despite the clear necessity demonstrated by the research previously cited in this article.

Programs with Human Factors Courses Available. Of the 121 academic institutions with complete data available via a public website, 43 (or 36%) offered human factors courses within the department or as electives, but did not require human factors courses for program completion. These programs averaged 33.6 credit hours required for completion. The number of human factors credit hours available averaged 5.2. Carnegie Mellon University was not included in these averages due to a different method of measuring credits (144 for completion and 96 human factors credits available).

The human factors-related courses available were generally titled a variant of "Human-Computer Interaction," "Introduction to Human Factors," and "Interactive Systems." A few programs developed courses focused on security applications, specifically, Johns Hopkins University's "Human Factors in Information Security" ("Johns Hopkins University Information Security Institute," 2013), Marymount University's "Human Considerations in Cybersecurity" ("Marymount University Graduate Catalog," 2013), and The University of North Carolina, Charlotte's "Usable Security and Privacy" ("UNC Charlotte College of Computing and Informatics," 2013). These courses show a specific intent to prioritize human factors in the cybersecurity program.

Several programs offered specific human factors concentrations or tracks within their cybersecurity programs, or offered separate human factors programs within the same department. Arizona State University, for example, offers an "Arts, Media, and Engineering (AME)" program that "emphasizes research on the integration of the human physical experience with computation and digital media" ("Arizona State University School of Computing, Informatics, and Decision Systems Engineering," 2013). Carnegie Mellon University hosts a Human-Computer Interaction Institute within its School of Computer Science with over fifty courses offered ("Carnegi Mellon University CyLab Graduate Programs," 2013). Indiana University offers a Master of Science in Human-Computer Interaction Design ("Indiana University Bloomington School of Informatics and Computing," 2013). Finally, The University of North Carolina, Charlotte, offers an option for a concentration in Human-Computer Interaction within its College of Computing and Informatics ("UNC Charlotte College of Computing and Informatics," 2013).

These courses and offerings represent a recent increased interest in prioritizing human factors in cybersecurity education, though they simultaneously show that the gap between the two disciplines has not yet been bridged.

Programs Requiring Human Factors Courses. Of the 121 academic institutions with complete data available via a public website, 3 (or 2%) required human factors courses for program completion. These programs averaged 34.0 credit hours required for completion and 4.0 human factors credit hours required. These particular programs merit detailing individually as there are so few and they may present models for other academic institutions to study.

Bellevue University's Master of Science in Cybersecurity requires a three credit hour course titled "Human Aspects of Cybersecurity." The course content includes "human behavior and interaction, motivation and influence, and social engineering. Emphasis is on the human element of cyber incidents in relation to protecting information and technology assets." This course is particularly unique because it specifically applies the concepts to cybersecurity, as mentioned in the previous section. The course materials available online do not appear to include content that specifically addresses design usability of cybersecurity systems, but seem to focus

primarily on social engineering vulnerabilities and reducing the unintentional insider threat ("Bellevue University Cybersecurity Degree - Master of Science," 2013).

Norwich University's Master of Science in Information Security and Assurance requires a six credit hour course titled "Human Factors and Managing Risk," which exposes students to: "security awareness as a component of organizational culture; the process of crafting an information assurance message; ethical decision-making as a factor in security; social psychology and how behaviors influence the effectiveness of security activities; the use of employment practices and policies to support information security, and the creation of acceptable use and email policies." This course, like Bellevue University's, is specifically designed to address cybersecurity and is integrated with National Institute of Standards and Technology publications. The course description implies that both human interaction vulnerabilities and system design usability are discussed ("Norwich University Master of Science in Information Security & Assurance," 2013).

The University of Texas at El Paso's Master of Science in Information Technology (MSIT) requires a three credit hour course titled "Human-Computer Interaction" in which students learn about "models of user behavior and human information processing, models of interaction, interaction styles including direct manipulation, interface design and development methods, implementation issues, interface programming, evaluation methods, and human-computer interaction research methods." The unusual focus on human factors may be a result of the MIST program's broad approach to information technology, which includes *management* of information along with the technical means to do so ("University of Texas at El Paso Computer Science," 2013).

Each of these three programs takes a slightly different approach to the integration of human factors into a cybersecurity curriculum, but they have all recognized the need to prioritize these elements.

CONCLUSIONS AND RECOMMENDATIONS

At a minimum, any cybersecurity curricula should include usability, cognition and psychology, social and cultural influences, and research methods and user testing as addressed in the beginning of this article. Many of the required or optional human factors courses offered by the academic institutions in this study focused on one or two of these elements instead of taking a holistic approach to the subject. It is important to remember that despite a common computer science emphasis, human-computer interaction and usability are not the only elements that influence human decision-making when it comes to cybersecurity.

The 62% of programs that do not yet include human factors as a part of the cybersecurity curriculum can provide their students with a more robust cybersecurity education by adding a human factors requirement. Since, on average, these programs require the least number of credit hours, it should not greatly disrupt the curricula to add a three credit hour course on cybersecurity. Some of these schools may already have human factors courses available in departments other than that in which the cybsersecurity program is offered. For example, the University of Kansas has a masters program in Interaction Design available in its Department of Design ("KU Interaction Design (MA)," 2013). It would be cost-effective and easily implemented to include coursework from this program as an interdisciplinary requirement for the computer science degree. Academic institutions without any human factors curricula should study other institution's cybersecurity-specific coursework as outlined in the Major Findings section to develop their own materials and requirements.

The 36% of schools that offer cybersecurity programs already have human factors courses in place, and some incorporate these courses as electives. Again, these programs on average require fewer credit hours toward graduation than programs that require human factors coursework, so it would not be very disruptive to increase the course-load by three credit hours. These academic institutions might seriously consider making their established

human factors courses mandatory. Additionally, the courses should be reviewed to ensure that they include usability, cognition and psychology, social and cultural influences, and research methods and user testing, as mentioned earlier.

The 2% of programs that require dedicated human factors courses as graduation requirements should also review course content to ensure it includes usability, cognition and psychology, social and cultural influences, and research methods and user testing. The next step is to encourage students to conduct further research on human factors requirements and theory as it applies to cybersecurity, and to publish their work for other scholars' and practitioners' use.

Finally, the National Security Agency has announced its intent to support the National Initiative for Cybersecurity Education by creating a program for National Centers of Academic Excellence in Cyber Operations ("National Centers of Academic Excellence - Cyber Operations," 2012). This program, along with the National Centers of Academic Excellence in Information Assurance, and other national programs like it, should consider adding human factors coursework as a requirement for entry into the program. Setting a national standard would greatly encourage many academic institutions to prioritize human factors as a part of their curricula.

TABLE 1. ACADEMIC INSTITUTION SURVEY RESULTS

Academic Institution	Total Req	HF Req	HF Avail	Source
Air Force Institute of Technology	48	0	6	("Air Force Institute of Technology Cyber Operations Master's Program," 2013)
Arizona State University	30	0	9	("Arizona State University School of Computing, Informatics, and Decision Systems Engineering," 2013)
Auburn University	30	0	12	("Auburn University Bulletin ~ Volume 108," 2013)
Bellevue University	36	3	3	("Bellevue University Cybersecurity Degree - Master of Science," 2013)
Boston University	32	0	0	("Boston University MS in Computer Science with Specialization in Cyber Security," 2013)
Bowie State University	36	0	0	("Bowie State University Department of Computer Science," 2013)
Brigham Young University	30	0	0	("Brigham Young University Bulletin Graduate Catalog," 2013)
California State Polytechnic University, Pomona	18	0	0	("Cal Poly Pomona - The Center for Information Assurance," 2013)
California State University, Sacramento	30	0	3	("Sacramento State Center for Information Assurance and Security," 2013)

Academic Institution	Total Req	HF Req	HF Avail	Source
California State University, San Bernardino	20	0	4	("California State University San Bernardino Information Assurance & Security Management Cyber Security Center," 2013)
Capella University	48	0	6	("Capella University University Catalog," 2013)
Capitol College	36	0	3	("Capitol College 2013–2014 Catalog," 2013)
Carnegie Mellon University	144	0	96	("Carnegi Mellon University CyLab Graduate Programs," 2013)
Champlain College	36	0	0	("Chaplain College MS in Managing Innovation & IT," 2013)
Clark Atlanta University	30	0	0	("Clark Atlanta University Graduate Catalog," 2013)
Colorado Technical University	48	0	5	("Colorado Technical University Degree Programs," 2013)
Columbus State University	36	0	3	("Columbus State University Master of Science Applied Computer Science," 2013)
Dakota State University	33	0	0	("Dakota State University Master of Science in Information Assurance & Computer Security," 2013)
Davenport University	37	0	0	("Davenport University Master of Science Information Assurance, MSIA," 2013)
DePaul University	52	0	0	("DePaul University Computer, Information and Network Security," 2013)
Drexel University	45	0	0	("Drexel University MS in Cybersecurity," 2013)
East Carolina University	30	0	0	("East Carolina University MS in Computer Science," 2013)
East Stroudsburg University of Pennsylvania	30	0	0	("East Stroudsburg University Computer Science, M.S.," 2013)
Eastern Michigan University	30	0	0	("Eastern Michigan University Information Assurance Graduate Courses," 2013)
Fairleigh Dickinson University	30	0	3	("Fairleigh Dickinson University M.S. in Computer Science," 2013)
Ferris State University	33	0	0	("Ferris State University Master of Science in Information Security and Intelligence," 2013)
Florida A&M University	30	0	0	("Florida A&M Department of Computer and Information Sciences," 2013)
Florida State University	35	0	3	("The Florida State University Computer Science," 2013)
Fort Hays State University	30	0	0	("Fort Hays State University Academic Programs," 2013)

Academic Institution	Total Req	HF Req	HF Avail	Source
Fountainhead College of Technology	33	0	0	("Fountainhead College of Technology Network Security and Forensics," 2013)
George Mason University	36	0	0	("George Mason University MS Management of Secure Information Systems," 2013)
Georgetown University	24	0	0	("Georgetown University," 2013)
Hampton University	36	0	0	("Master of Science in Information Assurance Program," 2013)
Howard University	48	0	0	("Howard University Department of Systems and Computer Science," 2013)
Idaho State University	N/A			("Idaho State National Information Assurance Training and Education Center," 2013)
Illinois Institute of Technology	30	0	0	("IIT School of Applied Technology, Master of Cyber Forensics and Security," 2013)
Illinois State University	15	0	0	("Illinois State University School of Information Technology," 2013)
Indiana University	30	0	27	("Indiana University Bloomington School of Informatics and Computing," 2013)
Indiana University of Pennsylvania	48	0	N/A	("Indiana University of Pennsylvania Information Assurance Program," 2013)
Information Resources Management College	39	0	0	("National Defense University Catalogs," 2013)
Iowa State University	30	0	0	("Iowa State University Information Assurance Center," 2013)
Jacksonville State University	33	0	3	("Jacksonville State University College of Graduate Studies Bulletin," 2013)
James Madison University	33	0	0	("James Madison University Master's Degree in Computer Science, Concentration in Information Security," 2013)
Johns Hopkins University	30	0	3	("Johns Hopkins University Information Security Institute," 2013)
Kennesaw State University	12	0	0	("Kennesaw State University Center for Information Security Education," 2013)
Lewis University	38	0	0	("Lewis University Graduate Catalog 2013–2015," 2013)
Louisiana Tech University	30	0	0	("Louisiana Tech University Catalog 2013–2014," 2013)
Marymount University	36	0	1	("Marymount University Graduate Catalog," 2013)

Academic Institution	Total Req	HF Req	HF Avail	Source
Mercy College	30	0	0	("Mercy College Cybersecurity Masters Degree," 2013)
Metropolitan State University	34	0	4	("Metropolitan State University Computer Science (MS)," 2013)
Mississippi State University	25	0	3	("Critical Infrastructure Protection Center at Mississippi State University," 2013)
Missouri University of Science and Technology	31	0	9	("Metropolitan State University Computer Science (MS)," 2013)
National University	54	0	0	("National University Master of Science in Cyber Security and Information Assurance," 2013)
Naval Postgraduate School	30	0	1	("Naval Postgraduate School Center for Information Systems Security Studies and Research," 2013)
New Jersey City University	36	0	0	("New Jersey City University Master of Science in National Security Studies," 2013)
New Jersey Institute of Technology	30	0	0	("New Jersey Institute of Technology MS in Cyber Security and Privacy," 2013)
New Mexico Tech	27	0	0	("New Mexico Tech Department of Computer Science & Engineering," 2013)
Norfolk State University	33	0	0	("Norfolk State University College of Science, Engineering, and Technology," 2013)
North Carolina A&T State University	30	0	6	("North Carolina Agricultural & Technical State Universtiy Master of Science in Computer Science," 2013)
Northeastern University	32	0	3	("Northeastern University College of Computer and Information Science M.S. in Information Assurance," 2013)
Norwich University	36	6	6	("Norwich University Master of Science in Information Security & Assurance," 2013)
NOVA Southeastern University	36	0	7	("NOVA Southeastern University Graduate School of Computer and Information Sciences Graduate Catalog," 2013)
Ohio State University	30	0	0	("Ohio State University Department of Computer Science and Engineering," 2013)
Oklahoma State University	33	0	0	("Oklahoma State University Center for Telecommunications & Network Security," 2013)
Our Lady of the Lake University	30	0	0	("Our Lady of the Lake University MS Information Systems and Security," 2013)
Pace University	36	0	3	("Pace University MS in Information Technology," 2013)

Academic Institution	Total Req	HF Req	HF Avail	Source
Polytechnic University	30	0	3	("New York University Polytechnic School of Engineering Master of Science Cybersecurity," 2013)
Polytechnic University of Puerto Rico	33	0	1	("Universidad Politécnica Puerto Rico M.S. Computer Science," 2013)
Regis University	36	0	0	("Regis University Master of Science in Information Assurance," 2013)
Rochester Institute of Technology	N/A			("Rochester Institute of Technology Center for the Advancement of Research and Education," 2013)
Rutgers, University	30	0	0	("Rutgers School of Arts and Sciences Computer Science," 2013)
Southern Illinois University Carbondale	N/A			("Southern Illinois University School of Information Systems and Applied Technologies," 2013)
Southern Methodist University	30	0	1	("Southern Methodist University M.S. Security Engineering," 2013)
Southern Polytechnic State University	36	0	0	("Southern Polytechnic State Institute M.S. Information Technology," 2013)
St. Cloud State University	30	0	0	("St. Cloud University Graduate Admissions," 2013)
State University of New York, Buffalo	30	0	0	("State University of New York, Buffalo Graduate Student Handbook," 2013)
Stevens Institute of Technology	33	0	0	("Stevens Institute of Technology Cybersecurity Graduate Program," 2013)
Syracuse University	30	0	0	("Syracuse University Master of Science of Computer Science," 2013)
Texas A&M - San Antonio	36	0	0	("Texas A&M University San Antonio Master of Business Administration," 2013)
Texas A&M University	21	0	2	("Texas A&M University Center for Information Assurance and Security," 2013)
Texas A&M University-Corpus Christi	36	0	1	("Texas A&M University-Corpus Christi, Masters Degree Computer Science Program," 2013)
The George Washington University	N/A			("George Washington University," 2013)
The Pennsylvania State University	30	0	0	("PennState Master of Science in Computer Science and Engineering," 2013)

Academic Institution	Total Req	HF Req	HF Avail	Source
The University of Texas at Dallas	33	0	0	("University of Texas at Dallas Cyber Security Research and Education Center," 2013)
The University of the District of Columbia	30	0	1	("University of the District of Columbia Department of Computer Science & Information Technology," 2013)
Towson University	33	0	0	("Towson University Applied Information Technology (M.S.)," 2013)
Tuskegee University	30	0	0	("Tuskegee University Master of Science in Information Systems & Security Management," 2013)
U.S. Military Academy, West Point	N/A			("West Point Cyber Research Center," 2013)
United States Air Force Academy	N/A			("United States Air Force Academy Department of Computer Science," 2013)
United States Naval Academy	N/A			("United States Naval Academy," 2013)
University of Advancing Technology	36	0	12	("University of Advancing Technology Master of Science in Information Assurance," 2013)
University of Alabama Huntsville	21	0	0	("University of Alabama in Hunstville Information Systems Major," 2013)
University of Alaska Fairbanks	45	0	0	("University of Alaska Fairbanks," 2013)
University of Arizona, Tucson	45	0	0	("University of Arizona Cybersecurity Fellowship Program," 2013)
University of Arkansas at Little Rock	31	0	0	("University of Arkansas at Little Rock Department of Computer Science," 2013)
University of California at Davis	36	0	0	("UC Davis Computer Science Master's Degree Requirements," 2013)
University of Colorado, Colorado Springs	30	0	6	("University of Colorado, Colorado Springs, Master of Engineering - Focus in Information Assurance," 2013)
University of Dallas	30	0	0	("University of Dallas Satish & Yasmin Gupta College of Business," 2013)
University of Denver	36	0	0	("University of Denver Computer Science 2013–2014," 2013)
University of Detroit, Mercy	N/A			("University of Detroit Mercy Graduate and Professional Studies," 2013)
University of Houston	36	0	0	("University of Houston Technology Master of Science in Information System Security," 2013)

Academic Institution	Total Req	HF Req	HF Avail	Source
University of Idaho	30	0	0	("University of Idaho M.S. Computer Science," 2013)
University of Illinois at Springfield	32	0	6	("University of Illinois at Springfield Center for Systems Security and Information Assurance," 2013)
University of Illinois at Urbana-Champaign	32	0	6	("University of Illinois at Urbana-Champaign M.S. and Ph.D. Degree Requirements," 2013)
University of Kansas	30	0	0	("KU School of Engineering Electrical Engineering & Computer Science," 2013)
University of Maryland University College	36	0	0	("University of Maryland University College Master of Science in Cybersecurity," 2013)
University of Maryland, Baltimore County	30	0	0	("University of Maryland Baltimore County Master's in Professional Studies: Cybersecurity," 2013)
University of Massachusetts, Amherst	30	0	0	("University of Massachusetts, Amherst School of Computer Science," 2013)Amherst School of Computer Science," 2013
University of Memphis	34	0	2	("The University of Memphis Graduate Catalog," 2013)
University of Minnesota	31	0	0	("University of Minnesota Computer Science & Engineering," 2013)
University of Missouri - Columbia	30	0	0	("University of Missouri Computer Science & IT," 2013)
University of Nebraska at Omaha	33	0	3	("University of Nebraska Omaha Master of Science in IA," 2013)
University of Nevada, Las Vegas	30	0	6	("University of Nevada, Las Vegas Graduate Catalog," 2013)
University of New Mexico	32	0	0	("University of New Mexico Computer Science Master's Degrees," 2013)
University of North Carolina, Charlotte	30	0	6	("UNC Charlotte College of Computing and Informatics," 2013)
University of North Texas	37	0	1	("University of North Texas Computer Science and Engineering," 2013)
University of Pittsburgh	30	0	0	("University of Pittsburgh Department of Computer Science," 2013)
University of Rhode Island	30	0	0	("University of Rhode Island Master of Science in Computer Science," 2013)

Academic Institution	Total Req	HF Req	HF Avail	Source
University of South Alabama	36	0	0	("University of South Alabama Undergraduate/Graduate Bulletin 2013–2014," 2013)
University of South Carolina	30	0	1	("University of South Carolina Master of Science in Computer Science and Engineering," 2013)
University of Tennessee at Chattanooga	33	0	0	("University of Tennessee Chattanooga Center for Information Security and Assurance," 2013)
University of Texas at El Paso	30	3	3	("University of Texas at El Paso Computer Science," 2013)
University of Texas, San Antonio	33	0	0	("University of Texas at San Antonio 2013–2015 Graduate Catalog," 2013)
University of Tulsa	30	0	0	("University of Tulsa Master of Science in Computer Science," 2013)
University of Washington	49	0	6	("University of Washington Curriculum," 2013)
Walsh College	30	0	0	("Walsh College Master of Science Information Assurance," 2013)
West Chester University of Pennsylvania	33	0	0	("Computer Science at West Chester University," 2013)
West Virginia University	34	0	0	("West Virginia University Masters of Science in Computer Science," 2013)
Wilmington University	36	0	3	("Wilmington University Cybersecurity Education," 2013)

REFERENCES

2013 Data Breach Investigations Report. (2013). Retrieved January 24, 2013, from http://www.verizonenterprise.com/resources/reports/rp_data-breach-investigations-report-2013_en_xg.pdf

Air Force Institute of Technology Cyber Operations Master's Program. (2013). Retrieved December 31, 2013, from http://www.afit.edu/en/docs/CCR/CCRCyber%20Operations%20Master%20Program.pdf

Arizona State University School of Computing, Informatics, and Decision Systems Engineering. (2013). Retrieved 30 December, 2013, from http://cidse.engineering.asu.edu/forstudent/graduate/computer-science/

Auburn University Bulletin ~ Volume 108. (2013). Retrieved December 30, 2013, from http://bulletin.auburn.edu/

Bellevue University Cybersecurity Degree - Master of Science. (2013). Retrieved December 31, 2013, from http://www.bellevue.edu/degrees/graduate/cybersecurity-ms/major-requirements.aspx

Boston University MS in Computer Science with Specialization in Cyber Security. (2013). Retrieved December 31, 2013, from http://www.bu.edu/academics/grs/programs/computer-science/ms-cyber-security/

Bowen, B. M., Devarajan, R., & Stolfo, S. (2011). *Measuring the human factor of cyber security.* Paper presented at the Technologies for Homeland Security (HST), 2011 IEEE International Conference on.

Bowie State University Department of Computer Science. (2013). Retrieved December 30, 2013, from http://www.cs.bowiestate.edu/Academics/Master%20of%20adm%20requirements.html

Brigham Young University Bulletin Graduate Catalog. (2013). Retrieved December 31, 2013, from http://graduatestudies.byu.edu/sites/default/files/graduatestudies.byu.edu/files/files/catalog/current-catalog.pdf

Cal Poly Pomona - The Center for Information Assurance. (2013). Retrieved December 30, 2013, from http://www.thecenteratcpp.com/

California State University San Bernardino Information Assurance & Security Management Cyber Security Center. (2013). Retrieved December 30, 2013, from http://iasm.csusb.edu/academic/graduate.html

Capella University University Catalog. (2013). Retrieved December 31, 2013, from http://www.capella.edu/assets/pdf/catalog.pdf

Capitol College 2013-214 Catalog. (2013). Retrieved December 30, 2013, from https://www.capitol-college.edu/files/file/PDFs/catalog13-14.pdf

Carnegi Mellon University CyLab Graduate Programs. (2013). Retrieved December 31, 2013, from https://www.cylab.cmu.edu/education/index.html

Centers of Academic Excellence Institutions. (2009). Retrieved January 4, 2014, from http://www.nsa.gov/ia/academic_outreach/nat_cae/institutions.shtml

Chaplain College MS in Managing Innovation & IT. (2013). Retrieved December 31, 2013, from http://www.champlain.edu/information-technology/managing-innovation-and-technology-masters/curriculum

Clark Atlanta University Graduate Catalog. (2013). Retrieved December 30, 2013, from http://www.cau.edu/CMFiles/Docs/OPAR/GRADUATE%20CATALOG%202010-2012%2003-12.pdf

Colorado Technical University Degree Programs. (2013). Retrieved December 30, 2013, from http://catalog.careered.com/~/media/Catalogs/ctu_6/course_catalog.pdf

Columbus State University Master of Science Applied Computer Science. (2013). Retrieved December 30, 2013, from http://academics.columbusstate.edu/catalogs/current/reqs/cobcs_msapcompsci.php

Computer Science at West Chester University. (2013). Retrieved December 31, 2013, from http://www.cs.wcupa.edu/grad/masters.html

Cranor, L. F. (2007). *Security and usability: Designing secure systems that people can use:* O'Reilly.

Critical Infrastructure Protection Center at Mississippi State University. (2013). Retrieved December 31, 2013, from http://www.security.cse.msstate.edu/cipc/

Dakota State University Master of Science in Information Assurance & Computer Security. (2013). Retrieved December 31, 2013, from http://www.dsu.edu/msia/

Davenport University Master of Science Information Assurance, MSIA. (2013). Retrieved December 31, 2013, from http://davenport.edu/programs/technology/master-of-science/information-assurance-msia

Definition and Domains of Ergonomics. (2013). Retrieved January 4, 2014, from http://www.iea.cc/whats/index.html

DePaul University Computer, Information and Network Security. (2013). Retrieved December 30, 2013, from http://www.cdm.depaul.edu/academics/Pages/2014/Requirements-MS-CINS-Network-Security.aspx

Drexel University MS in Cybersecurity. (2013). Retrieved December 31, 2013, from http://drexel.edu/ece/academics/grad/ms/cybr/

East Carolina University MS in Computer Science. (2013). Retrieved December 31, 2013, from http://www.ecu.edu/cs-acad/grcat/programCSCI.cfm

East Stroudsburg University Computer Science, M.S. (2013). Retrieved December 31, 2013, from http://www4.esu.edu/academics/catalog/graduate/computer-science-ms.cfm

Eastern Michigan University Information Assurance Graduate Courses. (2013). Retrieved December 31, 2013, from http://www.emich.edu/ia/graduate_courses.html

Fairleigh Dickinson University M.S. in Computer Science. (2013). Retrieved December 31, 2013, from http://view.fdu.edu/default.aspx?id=5977

Ferris State University Master of Science in Information Security and Intelligence. (2013). Retrieved December 31, 2013, from http://www.ferris.edu/business/documents/mba-misi/degrees/ISI-MISI-BU201208White_MSInfoSecurityIntelligence.pdf

Florida A&M Department of Computer and Information Sciences. (2013). Retrieved December 30, 2013, from http://www.famu.edu/index.cfm?cis&MastersinSE

The Florida State University Computer Science. (2013). Retrieved December 30, 2013, from http://www.cs.fsu.edu/current/grad/cs_ms.php

Fort Hays State University Academic Programs. (2013). Retrieved December 30, 2013, from http://www.fhsu.edu/informatics/academic-programs/

Fountainhead College of Technology Network Security and Forensics. (2013). Retrieved December 31, 2013, from http://fountainheadcollege.edu/technical-education-career-training-programs/areas-of-study/network-security-and-forensics

George Mason University MS Management of Secure Information Systems. (2013). Retrieved December 31, 2013, from http://som.gmu.edu/cyber-security-degree/courses/courses-syllabi/

George Washington University. (2013). Retrieved December 31, 2013, from http://www.gwu.edu/

Georgetown University. (2013). Retrieved December 31, 2013, from http://courses.georgetown.edu/index.cfm?Action=List&ProgramID=16

Gonzalez, J. J., & Sawicka, A. (2002). *A framework for human factors in information security.* Paper presented at the WSEAS International Conference on Information Security, Rio de Janeiro.

Howard University Department of Systems and Computer Science. (2013). Retrieved December 31, 2013, from http://www.scs.howard.edu/ms_requirements

Idaho State National Information Assurance Training and Education Center. (2013). Retrieved December 31, 2013, from http://www.niatec.info/ViewPage.aspx?id=0

IIT School of Applied Technology, Master of Cyber Forensics and Security. (2013). Retrieved December 30, 2013, from http://www.itm.iit.edu/cybersecurity/Program.php

Illinois State University School of Information Technology. (2013). Retrieved December 30, 2013, from http://it.illinoisstate.edu/graduate/admission/certificates/curriculum.shtml#IASeq

Indiana University Bloomington School of Informatics and Computing. (2013). Retrieved December 30, 2013, from http://www.soic.indiana.edu/graduate/programs/comp-sci/ms-requirements.shtml

Indiana University of Pennsylvania Information Assurance Program. (2013). Retrieved December 31, 2013, from http://www.cosc.iup.edu/infosecurity/IAProgram/BSIA.html

Iowa State University Information Assurance Center. (2013). Retrieved December 30, 2013, from http://www.iac.iastate.edu/content/node/64

Jacksonville State University College of Graduate Studies Bulletin. (2013). Retrieved December 30, 2013, from http://www.jsu.edu/graduate/pdf/2013-14Bulletin.pdf

James Madison University Master's Degree in Computer Science, Concentration in Information Security. (2013). Retrieved December 31, 2013, from http://www.infosec.jmu.edu/cohort2013.html

Johns Hopkins University Information Security Institute. (2013). Retrieved December 30, 2013, from http://isi.jhu.edu/mssi/program_requirements

Kennesaw State University Center for Information Security Education. (2013). Retrieved December 30, 2013, from http://infosec.kennesaw.edu/education.html#gradcert

Kissel, R. (2013). *NISTIR 7298 Revision 2: Glossary of Key Information Security Terms.* U.S. Department of Commerce Retrieved from http://nvlpubs.nist.gov/nistpubs/ir/2013/NIST.IR.7298r2.pdf.

Kraemer, S., Carayon, P., & Clem, J. (2009). Human and organizational factors in computer and information security: pathways to vulnerabilities. *computers & security, 28*(7), 509-520. doi: 10.1016/j.cose.2009.04.006

KU Interaction Design (MA). (2013). Retrieved December 30, 2013, from https://design.drupal.ku.edu/ma-interaction-design

KU School of Engineering Electrical Engineering & Computer Science. (2013). Retrieved December 30, 2013, from http://www.eecs.ku.edu/prospective_students/graduate/masters#computer_science

Lewis University Graduate Catalog 2013-2015. (2013). Retrieved December 30, 2013, from http://lewisu.smartcatalogiq.com/Graduate-2013-2015/graduate-catalog/College-of-Arts-and-Sciences/Department-of-Mathematics-and-Computer-Sciences/Information-Security-MS-Technical-Concentration

Louisiana Tech University Catalog 2013-2014. (2013). Retrieved December 30, 2013, from http://www.latech.edu/registrar/bulletin/louisiana_tech_university_catalog_2013-2014.pdf

Maguire, M. (2001). Methods to support human-centred design. *International Journal of Human-Computer Studies*(55), 587-634. doi: 10.1006/ijhc.2001.0503

Marymount University Graduate Catalog. (2013). Retrieved December 31, 2013, from http://marymount.edu/Media/Website%20Resources/catalog/2013/graduate/human-resources.htm#o8215

Master of Science in Information Assurance Program. (2013). Retrieved December 31, 2013, from http://science.hamptonu.edu/compsci/docs/ms_information_assurance.pdf

Mercy College Cybersecurity Masters Degree. (2013). Retrieved December 31, 2013, from https://www.mercy.edu/academics/school-of-liberal-arts/department-of-mathematics-and-cis/ms-in-information-assurance-and-security/

Metropolitan State University Computer Science (MS). (2013). Retrieved December 31, 2013, from http://www.metrostate.edu/msweb/explore/catalog/grad/index.cfm?lvl=G&section=1&page_name=computer_science_ms.html#_Curriculum

National Centers of Academic Excellence. (2009). Retrieved December 30, 2013, from http://www.nsa.gov/ia/academic_outreach/nat_cae/index.shtml

National Centers of Academic Excellence - Cyber Operations. (2012). Retrieved December 30, 2013, from http://www.nsa.gov/academia/nat_cae_cyber_ops/index.shtml

National Defense University Catalogs. (2013). Retrieved December 31, 2013, from http://www.ndu.edu/aa/catalogs.cfm

National University Master of Science in Cyber Security and Information Assurance. (2013). Retrieved January 4, 2013, from http://www.nu.edu/OurPrograms/SchoolOfEngineeringAndTechnology/ComputerScienceAndInformationSystems/Programs/Master-of-Science-in-Cyber-Security-and-Information-Assuranc.html

Naval Postgraduate School Center for Information Systems Security Studies and Research. (2013). Retrieved December 30, 2013, from http://cisr.nps.edu/mstrack.html

New Jersey City University Master of Science in National Security Studies. (2013). Retrieved December 31, 2013, from http://web.njcu.edu/sites/profstudies/securitystudies/content/masters.asp

New Jersey Institute of Technology MS in Cyber Security and Privacy. (2013). Retrieved December 31, 2013, from http://cs.njit.edu/academics/graduate/mscsp.php

New Mexico Tech Department of Computer Science & Engineering. (2013). Retrieved December 31, 2013, from https://cs.nmt.edu/academics/degree-programs/cs-masters/

New York University Polytechnic School of Engineering Master of Science Cybersecurity. (2013). Retrieved December 31, 2013, from http://engineering.nyu.edu/academics/programs/cybersecurity-ms

Norfolk State University College of Science, Engineering, and Technology. (2013). Retrieved December 31, 2013, from https://www.nsu.edu/cset/csetgraduate/cs-grad/cs-ms-curriculum#IA

North Carolina Agricultural & Technical State Universtiy Master of Science in Computer Science. (2013). Retrieved December 31, 2013, from http://www.ncat.edu/academics/schools-colleges1/coe/comp/pdfs/csghbk.pdf

Northeastern University College of Computer and Information Science M.S. in Information Assurance. (2013). Retrieved December 31, 2013, from http://www.ccs.neu.edu/graduate/degree-programs/m-s-in-information-assurance/

Norwich University Master of Science in Information Security & Assurance. (2013). Retrieved December 31, 2013, from http://online.norwich.edu/degree-programs/masters/master-science-information-security-assurance/curriculum

NOVA Southeastern University Graduate School of Computer and Information Sciences Graduate Catalog. (2013). Retrieved December 30, 2013, from http://www.scis.nova.edu/documents/catalog.pdf

Nurse, J. R., Creese, S., Goldsmith, M., & Lamberts, K. (2011a). *Guidelines for usable cybersecurity: past and present.* Paper presented at the Cyberspace Safety and Security (CSS), 2011 Third International Workshop on.

Nurse, J. R., Creese, S., Goldsmith, M., & Lamberts, K. (2011b). *Trustworthy and effective communication of cybersecurity risks: A review.* Paper presented at the Socio-Technical Aspects in Security and Trust (STAST), 2011 1st Workshop on.

Ohio State University Department of Computer Science and Engineering. (2013). Retrieved December 31, 2013, from http://www.cse.ohio-state.edu/grad/ms.shtml

Oklahoma State University Center for Telecommunications & Network Security. (2013). Retrieved December 31, 2013, from http://ctans.okstate.edu/index.php/prospective-students/graduate

Our Lady of the Lake University MS Information Systems and Security. (2013). Retrieved December 31, 2013, from http://www.ollusa.edu/s/1190/ollu-3-column-noads.aspx?sid=1190&gid=1&pgid=1420

Pace University MS in Information Technology. (2013). Retrieved December 31, 2013, from http://www.pace.edu/seidenberg/ms-in-internet-technology

PennState Master of Science in Computer Science and Engineering. (2013). Retrieved December 31, 2013, from http://www.cse.psu.edu/curriculum/graduate-handbook/fa13gradhandbook/masterofscience

Regis University Master of Science in Information Assurance. (2013). Retrieved January 4, 2014, from http://regis.edu/CPS/Academics/Degrees-and-Programs/Graduate-Programs/MS-Information-Assurance.aspx

Rochester Institute of Technology Center for the Advancement of Research and Education. (2013). Retrieved December 31, 2013, from http://care-ia.gccis.rit.edu/

Rutgers School of Arts and Sciences Computer Science. (2013). Retrieved December 31, 2013, from http://www.cs.rutgers.edu/graduate/ms_program.html

Sacramento State Center for Information Assurance and Security. (2013). Retrieved December 30, 2013, from http://www.ecs.csus.edu/csc/iac/cyberops.html

Southern Illinois University School of Information Systems and Applied Technologies. (2013). Retrieved December 31, 2013, from http://isat.siu.edu/

Southern Methodist University M.S. Security Engineering. (2013). Retrieved December 31, 2013, from http://www.lcsee.statler.wvu.edu/grad/masters-cs.php

Southern Polytechnic State Institute M.S. Information Technology. (2013). Retrieved December 30, 2013, from http://spsu.edu/itdegrees/degrees_and_certificates/MSIT-2013-14-Check-Sheet.pdf

St. Cloud University Graduate Admissions. (2013). Retrieved December 31, 2013, from http://www.stcloudstate.edu/gradadmissions/program/computer-science.aspx

State University of New York, Buffalo Graduate Student Handbook. (2013). Retrieved December 31, 2013, from http://www.cse.buffalo.edu/graduate/handbooks/grad-handbook-2013.pdf

Stevens Institute of Technology Cybersecurity Graduate Program. (2013). Retrieved December 31, 2013, from http://www.stevens.edu/sit/graduate/cybersecurity.cfm

Syracuse University Master of Science of Computer Science. (2013). Retrieved December 31, 2013, from http://lcs.syr.edu/our-departments/electrical-engineering-and-computer-science/academic-programs/masters/detail/computer-science

Texas A&M University-Corpus Christi, Masters Degree Computer Science Program. (2013). Retrieved December 31, 2013, from http://www.lcsee.statler.wvu.edu/grad/masters-cs.php

Texas A&M University Center for Information Assurance and Security. (2013). Retrieved December 31, 2013, from http://cias.tamu.edu/education

Texas A&M University San Antonio Master of Business Administration. (2013). Retrieved December 31, 2013, from http://www.tamusa.tamus.edu/collegeofbusiness/DegreeProgramsIndex/MBA.html

Theofanos, M. F., & Pfleeger, S. L. (2011). Shouldn't all security be usable? *Security & Privacy, IEEE, 9*(2), 12-17. doi: 10.1109/MSP.2011.30

Towson University Applied Information Technology (M.S.). (2013). Retrieved December 30, 2013, from http://grad.towson.edu/program/master/ait-ms/dr-ait-ms.asp

Tuskegee University Master of Science in Information Systems & Security Management. (2013). Retrieved 30 December, 2013, from http://www.tuskegee.edu/sites/www/Uploads/files/Academics/CBIS/computer_science/Courses%20Descriptions.pdf

UC Davis Computer Science Master's Degree Requirements. (2013). Retrieved December 30, 2013, from http://www.cs.ucdavis.edu/graduate/grad-req.html

UNC Charlotte College of Computing and Informatics. (2013). Retrieved December 31, 2013, from http://catalog.uncc.edu/sites/catalog.uncc.edu/files/media/Graduate-Catalogs/2013-2014-Grad-Catalog-16-COCI.pdf

United States Air Force Academy Department of Computer Science. (2013). Retrieved December 30, 2013, from http://www.usafa.edu/df/dfcs/

United States Naval Academy. (2013). Retrieved December 31, 2013, from http://www.usna.edu/homepage.php

Universidad Politécnica Puerto Rico M.S. Computer Science. (2013). Retrieved December 31, 2013, from http://www.pupr.edu/cs/ms_cs_graduation.asp

University of Advancing Technology Master of Science in Information Assurance. (2013). Retrieved 30 December, 2013, from http://majors.uat.edu/Information-Assurance/

University of Alabama in Hunstville Information Systems Major. (2013). Retrieved December 30, 2013, from http://catalog.uah.edu/preview_program.php?catoid=11&poid=891&returnto=211

University of Alaska Fairbanks. (2013). Retrieved 30 December, 2013, from http://www.uaf.edu/catalog/current/programs/it_specialist.html

University of Arizona Cybersecurity Fellowship Program. (2013). Retrieved 30 December, 2013, from http://mis.eller.arizona.edu/AZSecure/curriculum.asp

University of Arkansas at Little Rock Department of Computer Science. (2013). Retrieved December 30, 2013, from http://ualr.edu/computerscience/prospective-students/programs/graduate-programs/

University of Colorado, Colorado Springs, Master of Engineering - Focus in Information Assurance. (2013). Retrieved December 30, 2013, from http://www.eas.uccs.edu/cs/meia_program_info.shtml

University of Dallas Satish & Yasmin Gupta College of Business. (2013). Retrieved December 31, 2013, from http://www.udallas.edu/cob/ms/courses.html

University of Denver Computer Science 2013-2014. (2013). Retrieved December 30, 2013, from http://www.du.edu/learn/graduates/degreeprograms/bulletins/compsci/index.html

University of Detroit Mercy Graduate and Professional Studies. (2013). Retrieved December 31, 2013, from http://www.udmercy.edu/apply/grad-students/index.htm

University of Houston Technology Master of Science in Information System Security. (2013). Retrieved December 31, 2013, from http://www.uh.edu/technology/programs/graduate/information-system-security/

University of Idaho M.S. Computer Science. (2013). Retrieved December 30, 2013, from http://www.uidaho.edu/engr/cs/ms-computer-science

University of Illinois at Springfield Center for Systems Security and Information Assurance. (2013). Retrieved December 31, 2013, from http://csc.uis.edu/center/

University of Illinois at Urbana-Champaign M.S. and Ph.D. Degree Requirements. (2013). Retrieved December 30, 2013, from http://cs.illinois.edu/courses

University of Maryland Baltimore County Master's in Professional Studies: Cybersecurity. (2013). Retrieved January 4, 2014, from http://www.umbc.edu/cyber/programmaster.html

University of Maryland University College Master of Science in Cybersecurity. (2013). Retrieved December 31, 2013, from http://www.umuc.edu/academic-programs/masters-degrees/cybersecurity.cfm

University of Massachusetts, Amherst School of Computer Science. (2013). Retrieved December 31, 2013, from https://www.cs.umass.edu/grads/ms-requirements

The University of Memphis Graduate Catalog. (2013). Retrieved December 31, 2013, from http://www.memphis.edu/gradcatalog/degreeprog/cas/comp.php

University of Minnesota Computer Science & Engineering. (2013). Retrieved December 31, 2013, from http://www.cs.umn.edu/academics/graduate/degrees/ms.php

University of Missouri Computer Science & IT. (2013). Retrieved December 31, 2013, from http://engineering.missouri.edu/cs/degree-programs/

University of Nebraska Omaha Master of Science in IA. (2013). Retrieved December 31, 2013, from http://si2.ist.unomaha.edu/?p=msia

University of Nevada, Las Vegas Graduate Catalog. (2013). Retrieved December 31, 2013, from http://catalog.unlv.edu/index.php?catoid=11

University of New Mexico Computer Science Master's Degrees. (2013). Retrieved December 31, 2013, from http://cs.unm.edu/academics/degrees/masters_degrees/

University of North Texas Computer Science and Engineering. (2013). Retrieved December 31, 2013, from http://www.cse.unt.edu/site/node/53

University of Pittsburgh Department of Computer Science. (2013). Retrieved December 31, 2013, from http://www.cs.pitt.edu/grad/regulations_pages.php#4

University of Rhode Island Master of Science in Computer Science. (2013). Retrieved December 31, 2013, from http://www.cs.uri.edu/academics/graduate-studies/master-of-science-in-computer-science/

University of South Alabama Undergraduate/Graduate Bulletin 2013-2014. (2013). Retrieved December 30, 2013, from http://www.southalabama.edu/bulletin/cis.htm#mastercisis

University of South Carolina Master of Science in Computer Science and Engineering. (2013). Retrieved December 31, 2013, from https://www.cse.sc.edu/graduate/ms

University of Tennessee Chattanooga Center for Information Security and Assurance. (2013). Retrieved December 31, 2013, from http://www.utc.edu/center-information-security-assurance/academic-offerings.php

University of Texas at Dallas Cyber Security Research and Education Center. (2013). Retrieved December 31, 2013, from http://csrc.utdallas.edu/Education/Education.html

University of Texas at El Paso Computer Science. (2013). Retrieved December 31, 2013, from http://www.cs.utep.edu/grad/msit/DegreeRequirements.html

University of Texas at San Antonio 2013-2015 Graduate Catalog. (2013). Retrieved December 31, 2013, from http://www.utsa.edu/gcat/chapter6/COB/istmdept.html#msitiac

University of the District of Columbia Department of Computer Science & Information Technology. (2013). Retrieved December 31, 2013, from http://csit.udc.edu/graduate/courselistings.php

University of Tulsa Master of Science in Computer Science. (2013). Retrieved December 31, 2013, from http://www.utulsa.edu/academics/colleges/college-of-engineering-and-natural-sciences/departments-and-schools/tandy-school-of-computer-science/CS%20Programs%20of%20Study/graduate-and-professional-programs/MS-in-Computer-Science.aspx

University of Washington Curriculum. (2013). Retrieved December 31, 2013, from http://www.uwb.edu/cybersecurity/curriculum

Walsh College Master of Science Information Assurance. (2013). Retrieved December 31, 2013, from http://www.walshcollege.edu/informationassurancemastersdegree

West Point Cyber Research Center. (2013). Retrieved December 31, 2013, from http://www.westpoint.edu/crc/SitePages/Home.aspx

West, R. (2008). The psychology of security. *Communications of the ACM, 51*(4), 34-40. doi: 10.1145/1330311.1330320

West Virginia University Masters of Science in Computer Science. (2013). Retrieved December 31, 2013, from http://www.lcsee.statler.wvu.edu/grad/masters-cs.php

Wilmington University Cybersecurity Education. (2013). Retrieved December 31, 2013, from http://cae.wucot.org/

AUTHOR

Shana Kayne Beach (shana.beach@gmail.com) graduated from the University of Kansas in 2007 with a BFA in industrial design and an MA in communication studies. Upon graduation, she earned a commission into the United States Air Force through the Reserve Officer Training Corps. She currently holds Network+, Security+, and Certified Ethical Hacker certifications.

Extraction and Reasoning over Network Data to Detect Novel Cyber Attacks

Jim Jones, PhD | Carl Beisel

ABSTRACT

Detection of novel cyber attacks in real time is difficult due to (i) the large volume of data available, and (ii) an uncertain relationship between raw network data and novel attacks. We present an approach and experimental results addressing these challenges. We define *indirect-effect* observables that represent anomalies and other second order effects that frequently result from the basic elements of a cyber attack, but which also can occur under normal circumstances. We extract these observables from network data using deep packet inspection tools, then provide the observables as processed input to a Bayesian network. The Bayesian network is based on the basic steps required to execute a cyber attack, and the parameters of the Bayesian network are derived from input by subject matter experts. The Bayesian network reasons over the observables in combination, probabilistically associating groups of observables with the required high-level steps that any cyber attack must execute. Our approach reduces large volumes of network data to an evidence-based probability of an active cyber attack in real time. We demonstrate the effectiveness of our approach compared to a signature-based intrusion detection system using a custom exploit. Our models are explanatory, in that a user may examine the evidence and reasoning structure supporting an attack assessment; such a system will have value as a training, teaching, and evaluation aid beyond its value as an attack detection capability.

INTRODUCTION

Cyber attacks generally fall into one of two classes, distinguished by whether or not we have previously seen the tools and techniques used in the attack. Once the tools and techniques used in an attack have been observed, development of prevention and detection strategies is straightforward and usually effective against similar future attacks. This is the means by which most intrusion detection and intrusion preventions systems work. However, attacks which use tools and techniques which we have not seen before, i.e., *novel* attacks, generally go undetected for some amount of time during which significant damage may occur.

This research aims to detect novel attacks in real-time so that prevention, containment, and mitigation actions may be taken more quickly and potential damage minimized or avoided altogether. Our approach is based on the principle that certain actions must be taken by an attacker in order to execute a successful attack, and such actions create network observables. These observables are not unique to cyber attacks, but when reasoned over collectively can provide a reliable indicator of a cyber attack.

We have researched, implemented, and tested a system which extracts observables from network traffic and reasons over those observables in real time. Our system provides a dynamic assessment of cyber attack likelihood and presents explanatory information for use by a human operator or analyst. Our system effectively detected a novel attack which was not detected by Snort, a popular signature-based cyber attack detection tool.

Problem Statement

Existing cyber attack detection systems rely on knowledge of attack tools and techniques in order to develop attack prevention steps and detection signatures. However, some number of attacks employ tools and/or techniques which have not been previously observed. Such novel attacks are often called *0-day* attacks since the vulnerability and/or exploit employed has been known in the public domain for "zero days". Such novel attacks are rarely detected until well after the initial compromise has taken place. Often a novel attack is only detected when subsequent overt damage, such as a system shutdown, data destruction, or information exposure occurs. The damage inflicted by an attacker is directly proportional to the window of time between attack and detection, where these windows can range from minutes to years (Gorman, 2012).

For obvious reasons, we do not have solid data on the number of novel attacks which go undetected. However, we do have examples of damage caused by attacks which were novel at the time they occurred, such as the Stuxnet zero-day exploits, Conficker's polymorphism and frequent evolution to evade detection, or recent examples affecting Windows, Java, Acrobat Reader, and various gaming engines (Smith, 2013). Further, the quantity and pace of new antivirus detection signatures implies a robust supply of novel attack tools and techniques. For example, Symantec AV (2014) has almost 24 million signatures as of this writing, having added three million signatures in 2013 and six million in 2012.

Purpose

The purpose of this research is to develop scalable and practical real-time detection of novel cyber attacks which are currently undetected or only detected after overt damage has occurred. Our approach is designed to complement and integrate with existing cyber attack detection systems and to minimize false positives, thereby increasing the overall percentage of cyber attacks successfully detected while not unnecessarily increasing operator workload. Finally, we have implemented a user interface to explain the output and assessments provided by our approach in terms a cyber defense operator or analyst will understand, which serves as an operational, analytic, and training aid.

LITERATURE REVIEW

Cyber attack detection has been an active research area since at least the 1980s. Denning's (1987) early intrusion detection work on audit log processing and models of normal user behavior was followed closely by Haystack (Smaha, 1988) which was also based on logs and user behavior. In the 1990s, work shifted to detecting intrusions by looking at network traffic, now called *Network Intrusion Detection Systems (NIDS)*. In 1999, DARPA conducted a comparison of multiple host and network intrusion detection systems (Lippmann et al., 2000), in which NIDS performed rather poorly. The landscape of cyber attack detection and response now includes systems which take actions to stop active attacks (*Intrusion Prevention Systems, or IPS*), and both detection and prevention systems have network and host versions, so we have NIDS, NIPS, HIDS, and HIPS.

Modern intrusion detection systems are based on one or more of the following: signatures, behaviors, or anomalies. Signature approaches, such as Snort (Roesch, 1999), maintain byte patterns which appear in network traffic associated with a known cyber attack. Signature approaches are the oldest method and have an established track record of high accuracy for known attacks and poor detection of previously unseen attacks. Behavior approaches (noted above) are mostly limited to host-based intrusion detection, as modeling user behavior from network traffic is a difficult and unsolved challenge. Anomaly approaches use statistical or similar approaches to detect abnormal network traffic (Wang and Stolfo, 2004). While many cyber attacks, including novel ones, create network anomalies, anomaly detection systems have historically high false positive rates that have rendered them impractical.

Modeling attacker behavior and detecting when aspects of that realized has been an active research area since the 1990s (Kemmerer, 1997)(Valdes and Skinner, 2000)(Lippmann and Ingols, 2005). Aspects of our approach are similar to more recent work such as the Bayesian Network approach of (Xie et al., 2010), the effects aspects of Mitre's Indicators of Compromise (IOC) and HBGary's commercial Digital DNA product, and the anomaly basis of (Wang, 2006) and (Ippoliti and Zhou, 2012). However, our approach is unique in that we are reasoning over collections of anomalies to detect the necessary generic states and transitions required to execute an attack rather than detecting specific digital artifacts of an attack, i.e., signatures, detecting the specific actions of an attacker, i.e., behaviors, or triggering on individual or linear combinations of anomalies.

APPROACH

Our approach relies on the common, abstract, and essentially unavoidable steps required for any cyber attack against a networked host. We use finite state machine models of attack phases to generate evidence-driven reasoning models of generic attacks, where the evidence consists of observables extracted from network traffic. Evidence is collected and accrued to reasoning models in real time to confirm or deny attack hypotheses and thereby detect the presence of any attack, independent of attack details or signatures. We have dubbed the reasoning component of the system *HyReM* (Hybrid Reasoning Model), and the combined evidence collection and reasoning system *StORM* (Strategic Observation and Reasoning Model).

Cyber Attack Model

We consider a common class of network-based cyber attack where an attacker sends network traffic from one or more source systems to one or more target systems for the purposes of gaining and retaining control of one or more of the target systems. Much current attack activity, including nation-state espionage, advanced persistent threat insertion, financial account theft, and botnet construction falls into this class of attack. Such attacks are often executed using known vulnerabilities and exploits, i.e., an attacker probes a target to establish susceptibility to one or more known vulnerabilities, then selects an appropriate exploit and gains access to the target system. Publicly available tools such as Tenable Security's Nessus and Rapid7's MetaSploit implement such an approach with point-and-click simplicity and contain thousands of potential vulnerabilities and associated exploits.

Whether the tools and techniques used by an attacker are known in the cyber defensive community or not, and regardless of the attacker's intentions, most cyber attacks follow the same sequence of steps. In short, the attacker must (1) identify a target, (2) find a weakness in the target, (3) exploit that weakness to gain access, then (4) hide evidence of the attack and (5) ensure a means for continued access to the target. Such a five-phase attack sequence is common in the academic and other literature, see for example the detailed treatment by Skoudis and Liston (2005) in the book *Counter Hack Reloaded*. With few exceptions, a successful cyber attack must execute most or all of these steps, although steps may be combined and reordered. Steps are rarely omitted, since this leaves the attacker's effort incomplete and less likely to succeed or persist.

The five-phase model is suitable for conducting and analyzing most cyber attacks, since most cyber attackers use previously known vulnerabilities and exploits. However, our focus on novel attacks means that we are interested in the vulnerability discovery and exploit development phases of an attack as well, so we modified the common five-phase model to incorporate two preliminary steps, (i) vulnerability research and (ii) exploit development and testing. Our early research also indicated that the reconnaissance and scanning phases were essentially indistinguishable in our models, so we combined them into a single phase.

The result of our modifications is a six-phase model, where each phase represents an activity conducted by one or more attackers. In the lexicon of state machines, each of these phases represents

a *transition* from one *state* to another. Using the six phases as six transitions, we populated seven states and derived the state model in Figure 1. Such a model has three advantages for our research: (i) we may derive observables for states and/or transitions (our intuition is that they will be different), (ii) we may identify where in the model an active attack is, allowing accurate damage assessment and predictive defenses, and (iii) state diagrams are common in the cyber attack modeling literature, so we can later incorporate different cyber attack models into our research and development efforts. Our modeling approach does not require that attack steps are necessarily detected in sequence, nor that all steps are detected. Rather, our model is populated non-sequentially as evidence is accrued and our belief in an attack is updated in real time.

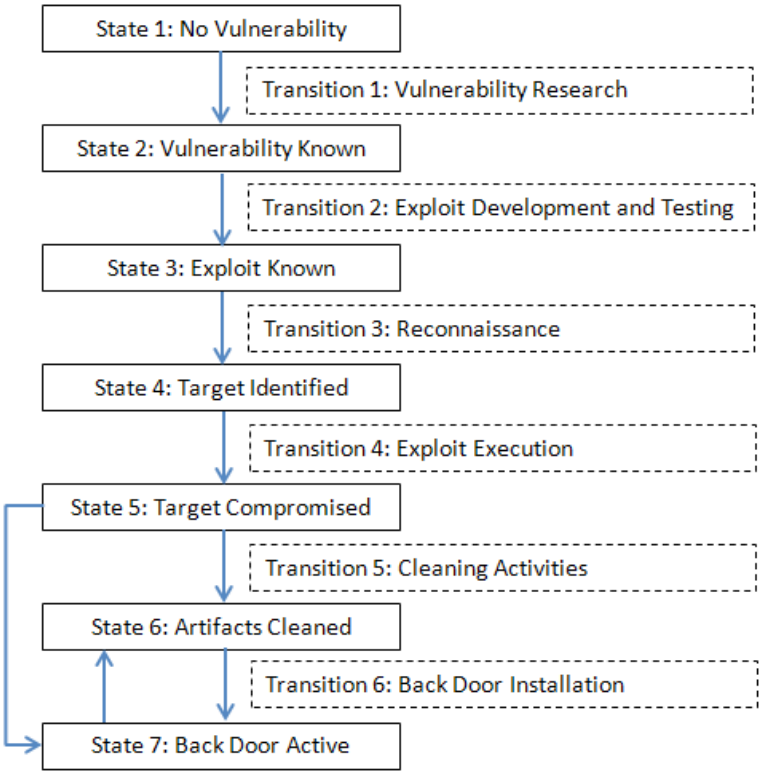

FIGURE 1. CYBER ATTACK STATE DIAGRAM

Observables

For the purposes of our work, we derived a set of network-based observables associated with the six state transitions. The observables, derived from discussions with subject matter experts and reviews of past attacks, represent anomalies or packet patterns that would likely occur during each transition, but which also may occur under other circumstances.

We deliberately avoided attack-specific observables, instead seeking more general and second order observables. As a result, our observable definitions do not require updates as attacker tools and techniques change and evolve.

In the list that follows, each observable is associated with a transition as indicated by the identifier (T1n = transition 1 = vulnerability research, T2n

= transition 2 = exploit development and testing, etc.). In most cases, a transition has more than one observable, indicated by a suffix letter in the identifier (a, b, c, ...). There is no significance to the ordering of observables for a given transition. For each observable, the list below provides the logic of the observable definition, the justification for the observable, and a reference if appropriate. For example, consider observable T1a where we are looking for observables associated with vulnerability research. A client (the potential attacker) might send a corrupted or otherwise unusual packet to a web server in an attempt to discover a vulnerability. Most web server responses under normal circumstances will contain an html tag. When presented with an unusual packet from the client, the server might respond abnormally with a packet that does not contain an html tag. As noted previously, this may occur under non-attack circumstances as well, for example when the server response is fragmented over multiple packets, or when an application uses port 80 with a protocol other than HTTP or to pass content other than HTML. Taken alone, a single T1a occurrence means very little. It is the probabilistic combination of observables, performed by the reasoning engine, that makes sense of multiple observables.

Most observable logic specifies "after 3-way handshake." This is the series of three packets used to establish a TCP connection. For our purposes, we are usually only interested in connections once they have been established. Additionally, many observable definitions are specifically for the server or client side of a TCP connection. In these cases, source or destination ports are used to distinguish the direction of the traffic.

T1a: After 3-way handshake, SrcPort=80, payload does not contain "<html>"
Justification: Abnormal response from web server

T1b: After 3-way handshake, SrcPort=80, payload="Bad Request"
Justification: Client sent unrecognized command to web server

T1c: After 3-way handshake, SrcPort=25, payload≠<ASCII>
Justification: Abnormal response from mail server (normally we expect plaintext command responses)

T1d: After 3-way handshake, SrcPort=25, payload="command unrecognized"
Justification: Client sent unrecognized command to mail server

T1e: After 3-way handshake, DstPort=80, payload≠<ASCII>
Justification: Abnormal traffic to web server (usually GET or POST with ASCII data)

T1f: After 3-way handshake, DstPort=25, payload≠<ASCII>
Justification: Abnormal traffic to mail server (usually EHLO, MAIL FROM, or RCPT TO with ASCII data)

T2a: Server does not send ACK after data packet from client
Justification: Client sent traffic that corrupted server and/or application; culprit could be current packet or one prior

T2b: Server does not send SYN-ACK for subsequent connection request from any client
Justification: Server (app or system) crashed

T3a: After 3-way handshake, DstPort=*, client sends FIN
Justification: Client closes connection after 3-way handshake without any data transfer

T3b: After 3-way handshake and one ASCII data transfer server to client, DstPort=*, client sends FIN
Justification: Client closes connection after 3-way handshake and banner, without any other data transfer

T3c: Server sends RST-ACK on closed port
Justification: Server is responding to a possible probe looking for open ports

T4a: Client to server traffic containing "\xeb\x21" followed within 100 bytes by \xe8\xda\xff\xff\xff"
Justification: Look for known shellcode characters/ patterns; this is a simple example

Reference: http://www.net-workforensics.com/2010/05/16/ network-detection-of-x86-buffer-overflow-shellcode/

T4b: Client payload contains 5+ identical and consecutive NOP instruction byte patterns
Justification: A "NOP sled" is a common technique used in buffer overflow exploits; the sled consists of multiple NOP (No Operation) instructions to ensure that the real instructions fall in the desired range

T5a: Client to server traffic where port≠23 and first 100 bytes of payload contains "rm", "rmdir", "rd", "del", or "erase"
Justification: File or directory removal activity; very loose rule; lots of false positives, but useful when combined with others

T6a: First two bytes of client to server payload="MZ"
Justification: COM, DLL, DRV, EXE, PIF, QTS, QTX, or SYS file transfer for use in a backdoor (will get some false positives as header is not unique)

Reference: http://www.garykessler.net/library/file_ sigs.html

T6b: New Port opened on server; learning mode first 500 packets
Justification: Traffic from a port not previously seen might indicate the opening of a new back door

T6c: Payload with recurring duplicate string of encrypted data are being transmitted
Justification: Attacker tools sometimes use poor cryptographic implementations; such implementations may re-use the same seed for different packets with a common header, resulting in some ciphertext repeating across packets

T6d: Unencrypted payload or payload missing SSL or SSH header on ports 22 or 443
Justification: Ports 22 and 443 normally carry encrypted traffic; unencrypted traffic on these ports might indicate an attacker using these ports for unauthorized communication

T6e: Payload with encrypted data found when port is not 22, 443, 993, 995, 585, 465, 3389
Justification: Encrypted data on a port not normally carrying encrypted data might indicate an active back door attempting to use ports commonly passed through a firewall

T6f: Client payload on port 80 with PHP tags
Justification: While PHP has many legitimate uses, it is also heavily leveraged by the attacking community

Taken individually or in linear combinations, our observables might be used in a signature-based or anomaly-based attack detection system. However, using our general observables in such a fashion would result in significant false positives and would be impractical. It is our reasoning model, discussed in the next section, that processes these observables into reliable assessments of cyber attacks.

Reasoning Model

The inspiration for our reasoning model comes from human cyber attack analysts. From the authors' experience and discussions with other analysts, we know that novel attack detection begins with an anomaly. From that starting point, the analyst looks for other evidence to support or refute a working hypothesis that an attack is underway. Assuming the analyst accumulates supporting evidence, their assessment of attack likelihood increases. At some point, the analyst determines that a threshold of support and belief has been crossed and the alert is passed on for action. Justification for the alert is a critical element of the analyst's task, as they must explain why they believe an attack is occurring or has occurred. In other cases, additional supporting evidence is not present and the analyst disregards the anomaly.

Bayesian networks (Pearl, 1988) are probabilistic reasoning structures capable of reasoning over uncertain and incomplete evidence. Bayesian networks are widely used for reasoning problems where we need to represent expert human knowledge and apply that expert knowledge to future evidence and situations. Bayesian networks are made up of nodes and links (i.e., a mathematical graph), where nodes represent evidence and hypotheses. Each node contains a *conditional probability table (CPT)* which represents the influence of connected nodes on that node. It is both the structure (the number of nodes and their connections) and the CPT which capture the expert prior knowledge. When new evidence is introduced, it changes the value of one or more nodes, and the effects of those changes are propagated throughout the network. In this manner, a set of evidence items (input at various nodes) affects the probability of an associated hypothesis (the end value at some other node).

Early applications of Bayesian networks included medical diagnosis. Experts were consulted to build the networks, which consisted primarily of symptoms and illnesses. When a patient presented with certain symptoms, those nodes in the network would be set, and the probability of various illnesses would be output as the values of other nodes. Our problem is similar, in that we are trying to capture the expert knowledge of what combination of evidence causes an analyst to believe a cyber attack is present.

Like medical diagnosis, we have uncertain and partial information. Unlike medical diagnosis, we do not have a pre-existing set of conditions, systems, and patient histories from which to construct our network. Therefore, we constructed a Bayesian network based on our attack model. We structured our network around our six-phase attack model, so a root node indicating probability of compromise is connected to six nodes, each representing a state transition. Each of the observables described above is connected to the appropriate state transition. This base model with subnets for web (port 80) and email (port 25) server attacks broken out in Transition 1 is shown in Figure 2.

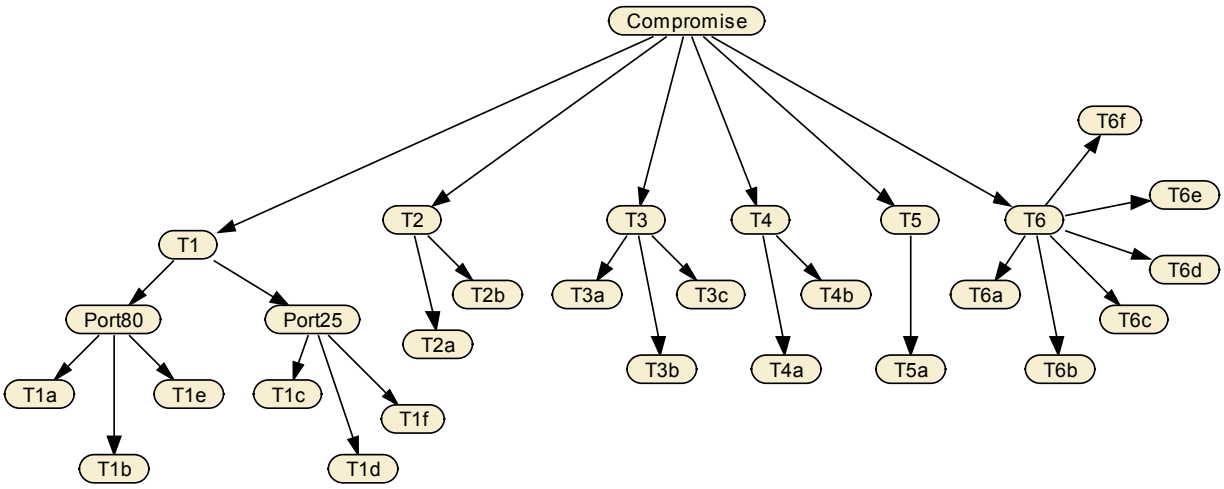

FIGURE 2. BASE BAYES NET MODEL

We iteratively refined the BN model structure and established the model parameters based on conversations with cyber attack and network subject matter experts and preliminary experiments using test data and traffic captured from live networks. Structural refinements included the addition of three multi-transition observables (M1, M2, and M3). M1 aggregates observables across Transitions 1-4 to represent attacker activity up to the point of a system compromise. M2 and M3 reflect observable concentrations to a specific target port or from a specific source IP address, respectively. The revised model is shown in Figure 3.

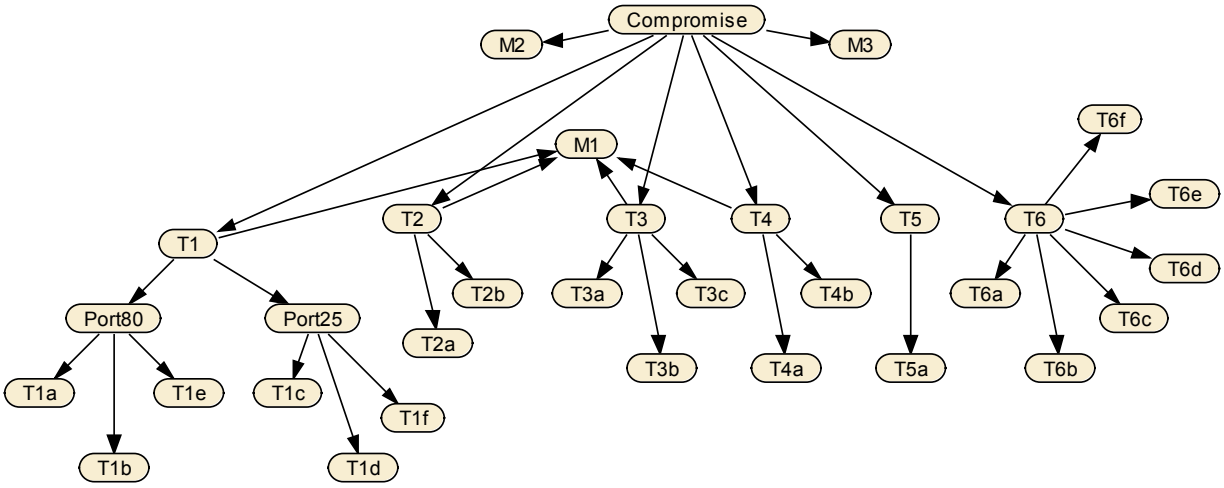

FIGURE 3. REFINED BAYES NET MODEL

The reasoning system ingests raw observables and processes them into likelihoods which are then assigned to the observable nodes in the BN model. This processing uses a scalable saturation curve to assign observable node values based on the number of respective observable instances reported after weighting for time. This curve has the shape shown in Figure 4 and is scaled differently for each observable to reflect that observable's background level, i.e., the prevalence of that observable in the absence of an attack. Scaling values for the observables were estimated from a limited analysis of laboratory-collected traffic samples known to not contain attack traffic.

We also applied a time decay function to observables (see Figure 5) prior to computing node values. In practice, most attacks occur in small time windows on the order of minutes or hours. However, some attacks are perpetrated over weeks or months specifically to avoid detection, aka the "low and slow" attack. Our exponential decay function weights observables more heavily when they are in close chronologic proximity but allows for some residual weight to persist indefinitely. In a sustained deployment, we would likely impose some arbitrary cutoff on observable impact, possibly after one year or so.

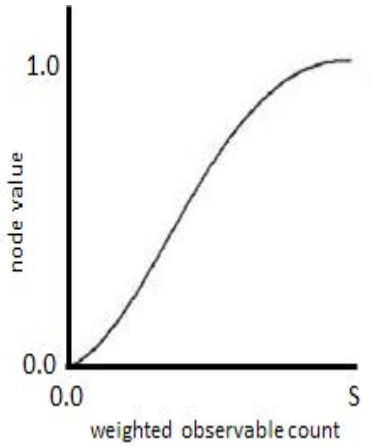

FIGURE 4. SATURATION CURVE

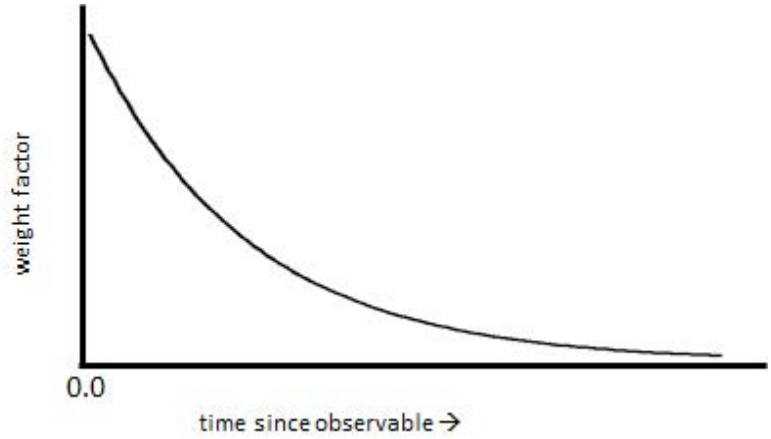

FIGURE 5. TIME DECAY CURVE

IMPLEMENTATION

We implemented our approach as shown in Figure 6. Observables are extracted from live network traffic (A) and sent to the reasoning engine via a TCP network connection and postgres database (B). The reasoning engine pre-processes the observables (C), then sets the appropriate node values in the Bayes Net (D). The Bayes Net outputs probability of compromise and supporting data to the human operator via a GUI (E).

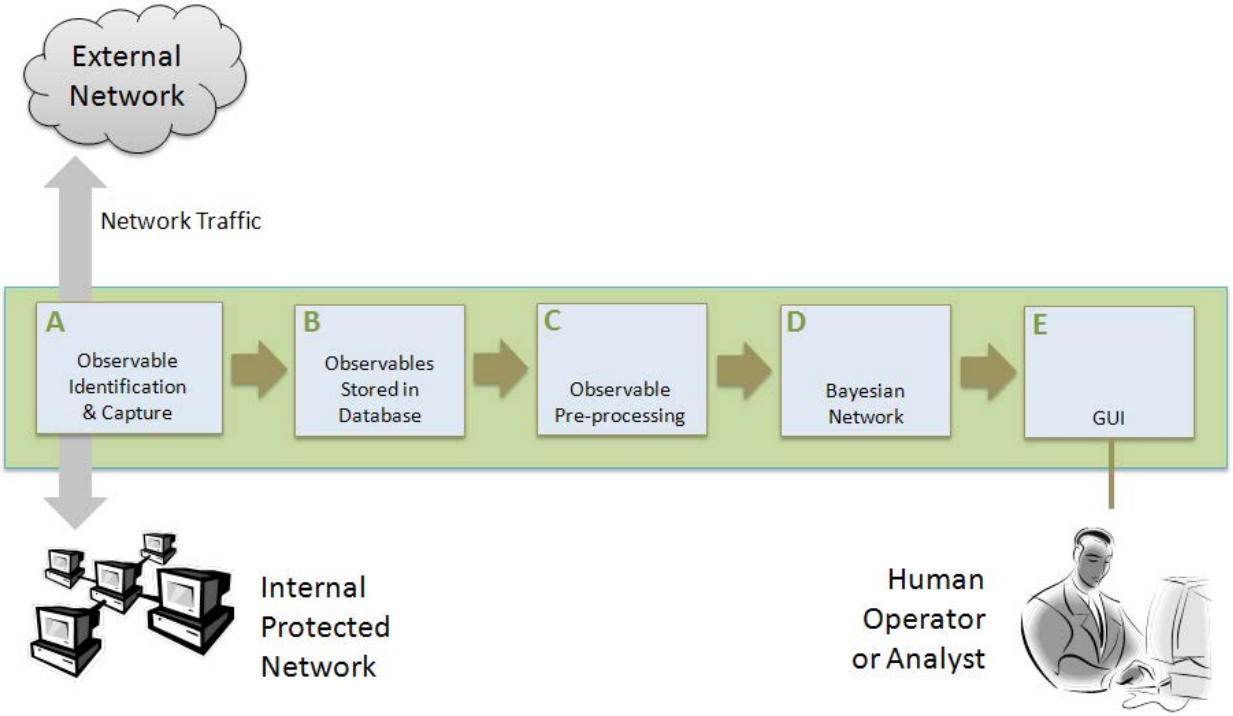

FIGURE 6. IMPLEMENTATION FUNCTIONAL DIAGRAM

We designed and implemented the system to be modular and scalable. Each component of the system may be replaced by an alternative implementation, and each component of the system may be distributed to facilitate scalability. For example, our system design and implementation supports multiple distributed observable capture units, a distributed observables database, and distributed or alternate reasoning model.

Observable Extraction

We implemented the observable definitions listed previously in CloudShield (PacketC code) and using the C++ libpcap library in a Linux (Ubuntu) environment. The two observable capture implementations are functionally identical, but CloudShield is a commercial product providing high performance (greater than 1 Gbps throughput) and libpcap is open source and free with lower throughput. In both cases, live network traffic or stored capture files are processed to generate observable strings which are then sent over a TCP socket to a listening agent on the reasoning server. Observable strings are comma-separated text strings with the following fields: timestamp, device ID, observable ID, attacker IP, attacker port, target IP, target port, and protocol.

Reasoning Model Processing

Observable strings are received by the reasoning server and stored in a local postgres database. The applications running on the reasoning server are written in Java and the Bayes Net implementation uses the open source UnBBayes Java classes from the University of Brazil (Matsumoto, 2011). The observable pre-processor polls the database at regular intervals or when prompted by the human user. The pre-processor derives node values from the stored observables and updates the Bayes Net node values accordingly. Our current implementation maintains a distinct Bayes Net for each target (internal) IP address. The node changes are propagated through the networks, and updated information is stored in separate postgres tables for retrieval by the GUI.

User Interface

The user interface is implemented in Java and provides a view into the observable strings database, control over model updating, model output summaries, and a graphical drill-down capability for each model and primary subnodes (e.g., transition nodes). The interface provides summary information for all target IP addresses in the database, as shown in Figure 7 for a sample data set.

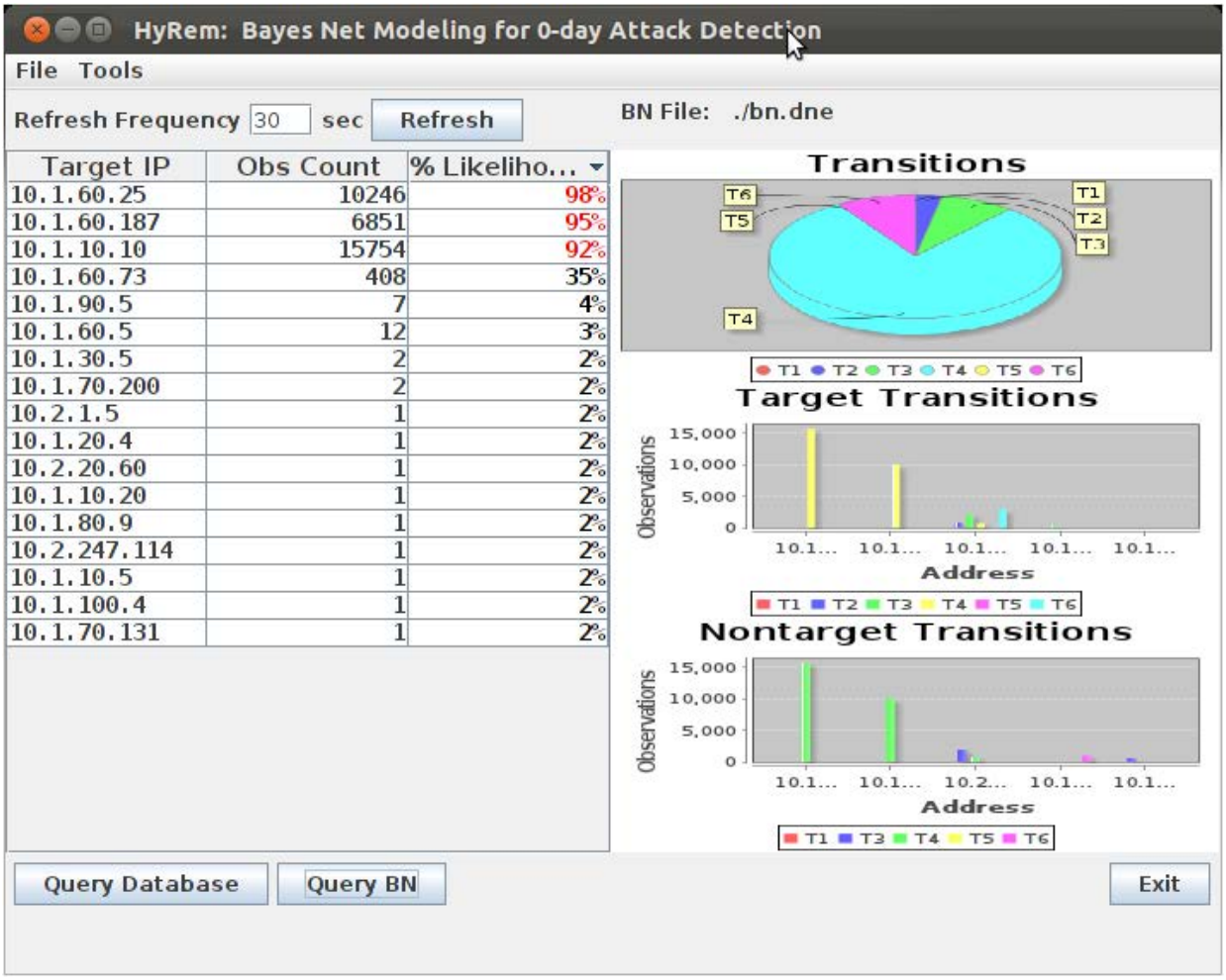

FIGURE 7. USER INTERFACE SUMMARY SCREEN

The summary screen sorts the IP addresses by decreasing likelihood of compromise and indicates the number of observables received for each IP address. The graphics on the right side of the display summarize all observables received for each transition and the most active target and nontarget (attacker) IP addresses. Selecting an IP address and clicking "Query Database" will present the observable strings associated with that IP address, and clicking "Query BN" will present a breakdown of transitions and observables supporting the likelihood of compromise value (see Figure 8 for a sample of this drilldown).

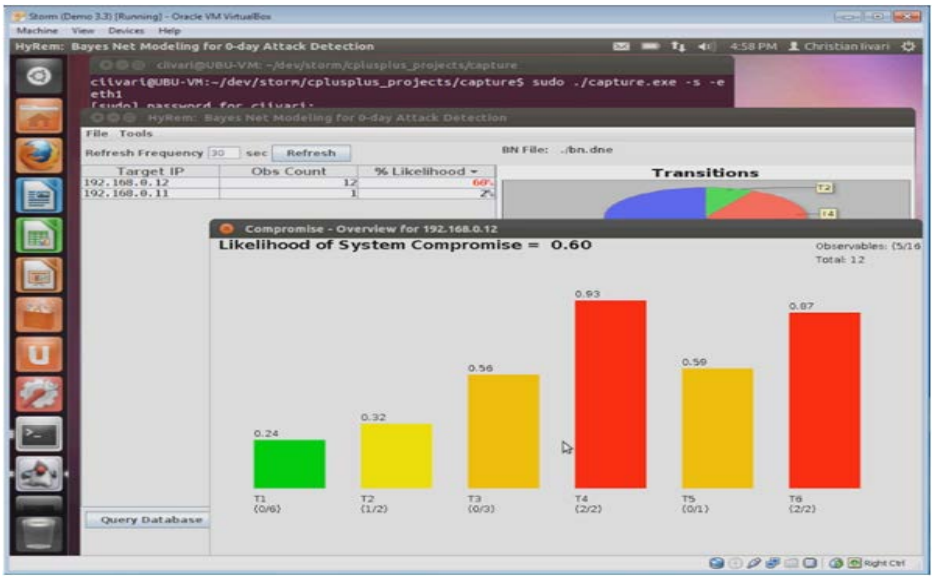

FIGURE 8. SAMPLE TARGET DRILLDOWN

In this display, the probability that each transition is present is shown as a numerical value on top of each bar. The bars are colored green, yellow, and red at arbitrary threshold values. The values in parentheses below each bar indicate the number of supporting observable types received out of the total possible types for each transition. For example, Transition 2 has two different supporting observables, and one of these has been detected. Clicking on any of the bars in the display of Figure 8 will present a similar breakdown for the observables supporting that transition.

EXPERIMENTAL PROCEDURES AND RESULTS

We established both virtual machine (VM) and portable environments to support testing, and we identified three scenarios to be tested: (1) attacks present with known ground truth, (2) attacks present with unknown ground truth, and (3) unknown attacks with unknown ground truth. We report on these experiments below.

Our Scenario 1 tests with known ground truth consisted of a virtual machine environment with four virtual machines running under VMWare workstation on a single computer: an attacker, a target, a monitor, and the StORM system. The attacker is a BackTrack5 (Ubuntu) system with the Armitage front end to MetaSploit. The target is an unpatched Windows XP sp1 system with the MiniShare application running. The monitor is another BackTrack5 system with Snort running. The StORM system is a full implementation of our approach using the libpcap-based observable capture application.

The goal of our initial testing was to establish whether or not StORM would detect a novel cyber attack, and to compare StORM to Snort, an existing signature-based attack detection tool. Prior to establishing the details of and conducting the tests, we locked down the configuration of the StORM system (including the BN model) to prevent attack-specific tuning. We devised two experiments: a noisy attack using known vulnerabilities and exploits, and a stealthy attack using known vulnerabilities but previously unseen exploits. In both cases, we started all four virtual machines from clean snapshot images, then started Snort and StORM on their respective virtual machines.

We used Metasploit on the attacker system to conduct the noisy attack (a video of the attack is available at http://www.xbit.cc/storm). The attack consisted of a port and vulnerability scan, successful

exploit execution (ms04_011_lsass), and remote command shell interaction. At each step in the attack, we observed the output of Snort (number and level of alerts) and StORM (number of evidence items and probability of compromise). After all stages of the attack, Snort reported 13 alerts (2 of which were Priority 1), and StORM captured 36 observables and assessed a likelihood of compromise of 92% (Test 1 in Table 1). Tests 2 and 3 in Table 1 used different vulnerabilities with the same attack steps and produced similar results. For each test, Table 1 records the raw number of packets on the network (or in the pcap file), the number of observables extracted from the packets, the ratio of observables to raw packets, and the probability of compromise, P(C), computed by the StORM system. Table 1 also summarizes Snort output where available on the same raw packet samples. The number of Snort alerts are totaled in the first Snort column and broken out by Priority in the other three Snort columns.

To conduct the stealthy attack experiments (video at http://www.xbit.cc/storm), we modified an existing exploit (http://www.securiteam.com/exploits/6X00B1PBPC.html) to emulate a 0-day attack. The exploit was originally written for Windows 2000 and takes advantage of a buffer overflow vulnerability in an older version of the MiniShare application. We separately verified that Snort does detect this particular exploit. We modified the exploit to run on Windows XP (a one-line change to a platform-specific address in the exploit code[1]), and ran an older version (1.4.1) of MiniShare on the target system. Tests 4 and 5 in Table 1 included reconnaissance activity prior to running the exploit and additional activity after running the exploit; the reconnaissance activity was detected by Snort (and StORM). For Test 6, we ran a simple reconnaissance check (telnet to port 80 on the target to verify that the port was active), ran the modified exploit, then executed post-attack activity: interacted with the exploit command shell, configured and started a back door using the telnet server on an arbitrary port, connected to the back door, transferred rootkit binaries via FTP, ran the rootkit to hide our backdoor process, and deleted our rootkit files. Snort reported 0 alerts during Test 6, while StORM reported a gradually increasing likelihood of compromise which peaked at 60%. We also ran a small sample of known clean traffic (Test 7), and combinations of prior network traffic captures (Tests 8-12).

TABLE 1. EXPERIMENT SUMMARY

			StORM			Snort			
Test #	Num Packets	Description	Num Obs	Obs/Raw	P(C)	Num Alerts	Priority 1	Priority 2	Priority 3
1	2179	Metasploit exploit ms04-011	36	2%	0.92	13	2	8	3
2	2588	Metasploit exploit ms03-026	48	2%	0.98	12	2	8	2
3	2420	Metasploit exploit ms08-067	47	2%	0.98	12	2	8	2
4	1669	Minishare - noisy	18	1%	0.95	13	2	8	3
5	1642	Minishare - moderate	16	1%	0.94	10	2	6	2
6	5650	Minishare - quiet	12	0%	0.60	0	0	0	0
7	325	Clean 1	1	0%	0.02	0	0	0	0
8	3848	Merge of tests 1 and 4	33	1%	0.99	23	4	16	3
9	2913	Merge of tests 2 and 6	50	2%	0.97	12	2	8	2
10	1994	Merge of tests 4 and 6	18	1%	0.95	13	2	8	3
11	6981	Merge of tests 5 and 7	51	1%	0.93	15	2	11	2
12	9428	Merge of tests 3, 4, and 7	96	1%	0.99	31	4	21	6
13	43659	Sensor file	540	1%	0.86				
14	371061	407-USMA	10246	3%	0.98	497			
15	335130	408-USMA	314	0%	0.77	354			

[1]*For the curious, the change is to replace #define RET "\xB8\x9E\xE3\x77" /*2k sp2*/ with #define RET "\x33\x55\xdc\x77" /*XP*/*

Scenario 2 tests, where we have an unknown ground truth but attacks are known to be present, are ongoing. Test 13 in Table 1 was run on a pcap (network capture) file provided from a production environment where an unauthorized attack was known to have occurred but we were not provided any additional information. StORM correctly identified the single target of the attack (probability of compromise = 0.86) from 286 systems with activity in the pcap file. Snort was not run on this dataset. Tests 14 and 15 were run on NSA packet captures #7 and #8 from the USMA 2009 CDX (Cyber Defense Exercise at the US Military Academy, available at https://www.itoc.usma.edu/research/dataset). CDX and similar Capture the Flag exercises are known to have attack traffic, but full ground truth is typically not available. We identified several systems that were likely compromised (only the highest P(C) is shown in Table 1), and a review of exercise details is underway to establish ground truth. We are also continuing to run the remaining pcap files from the CDX activity as well as other publicly available Capture the Flag and similar exercise data.

Scenario 3 tests, where we have no ground truth and no knowledge of underlying attack activity, are ongoing. We are collecting network data from enterprise networks, academic networks, home networks (with the owner's permission), and public networks where packet capture is authorized. Where packet retention is not authorized, we built a portable version of the StORM system for live, real-time processing of network traffic which only retains observable descriptions but no packet content. Both the lab (VM) and portable StORM implementations support replay of previously captured network traffic as well as real time processing of live network feeds.

CONCLUSIONS

We have demonstrated an approach to elicit useful cyber attack intelligence, i.e., the presence of a novel attack, in real time from a large volume of uncertain source data. Our approach reduces an input stream of network traffic by a factor of 100, i.e., roughly 1 in 100 original packets trigger an observable and require processing. Preliminary results indicate that this reduction will hold at scale (Tests 13-15 in Table 1). Each component of our design and implementation may be distributed to support scalability. Our prototype implementation successfully detected a novel attack which was not detected by existing methods.

Our reasoning model encodes expert knowledge, and our user interface is designed to explain the evidentiary basis for the system's assessment of compromise likelihood. In addition to production deployments for novel attack detection, the system has value as an educational and training tool. Students would be presented with the network evidence in real time or as a stored capture file and charged with identifying possible system compromises. The students' reasoning and conclusions could then be compared to StORM's output and model explanations. A similar activity with established experts may be used to further refine the models and performance of the StORM system.

Future work will initially focus on additional testing of previously captured network traffic (e.g., from capture the flag and similar exercises), production network data, and live network testing using our portable system implementation. Concurrently, we will be developing additional models to address new scenarios, e.g., coordinated data exfiltration across multiple systems, and we will be refining our current model. Our model as currently implemented is expected to detect some but not all novel attacks. Future model extensions will expand the set of novel attacks we can detect. Other future work will include a distributed implementation to demonstrate scalability, testing of alternative reasoning approaches, integration with existing systems for traffic redirection and system analysis, and use of the StORM system as a training platform for students and cyber attack analysts.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of Science Applications International Corporation (SAIC), and specifically wish to thank Robert Eek, Christian Iivari, and John Biddle for their significant contributions.

REFERENCES CITED

Denning, D. E. (1987). An intrusion-detection model. *Software Engineering, IEEE Transactions on, (2),* 222–232.

Gorman, S. (2012). Chinese hackers suspected in long-term Nortel breach. *The Wall Street Journal*, February 14, 2012. Retrieved from http://online.wsj.com/news/articles/SB10001424052970203363504577187502201577054.

Ippoliti, D., & Zhou, X. (2012). A-GHSOM: An adaptive growing hierarchical self organizing map for network anomaly detection. *Journal of Parallel and Distributed Computing*.

Kemmerer, R. A. (1997). NSTAT: a model-based real-time network intrusion detection system. Computer Science Department, University of California, Santa Barbara, Report TRCS97-18, http://www.cs.ucsb.edu/TRs/TRCS97-18.html.

Lippmann, R., Haines, J. W., Fried, D. J., Korba, J., & Das, K. (2000). The 1999 DARPA off-line intrusion detection evaluation. *Computer networks, 34(4)*, 579–595.

Lippmann, R. P., & Ingols, K. W. (2005). An annotated review of past papers on attack graphs (No. PR-IA-1). MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB.

Matsumoto, S., Carvalho, R. N., Ladeira, M., da Costa, P. C. G., Santos, L. L., Silva, D., ... & Cai, K. (2011). UnBBayes: a java framework for probabilistic models in AI. *Java in Academia and Research*. iConcept Press. http://unbbayes.sourceforge.net.

Pearl, J. (1988). Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann.

Roesch, M. (1999, November). Snort: Lightweight Intrusion Detection for Networks. In *LISA (Vol. 99*, pp. 229–238).

Skoudis, E., & Liston, T. (2005). Counter Hack Reloaded: A Step-by-Step Guide to Computer Attacks and Effective Defenses (Prentice Hall Series in Computer Networking and Distributed Systems).

Smaha, S. E. (1988, December). Haystack: An intrusion detection system. In Aerospace Computer Security Applications Conference, 1988., Fourth (pp. 37–44). IEEE.

Smith, A. D. (2013). 5 examples of zero-day attacks. *Network World*, August 12, 2013. Retrieved from http://www.networkworld.com/news/2013/081213-zero-day-examples-272672.html.

Symantec (2014). Virus Definitions & Security Updates. Retrieved from http://www.symantec.com/security_response/definitions.jsp).

Valdes, A., & Skinner, K. (2000, January). Adaptive, model-based monitoring for cyber attack detection. In *Recent Advances in Intrusion Detection* (pp. 80–93). Springer Berlin Heidelberg.

Wang, K., & Stolfo, S. J. (2004, January). Anomalous payload-based network intrusion detection. In *Recent Advances in Intrusion Detection* (pp. 203–222). Springer Berlin Heidelberg.

Wang, K. (2006). Network payload-based anomaly detection and content-based alert correlation (Doctoral dissertation, Columbia University).

Xie, P., Li, J. H., Ou, X., Liu, P., & Levy, R. (2010, June). Using Bayesian networks for cyber security analysis. In *Dependable Systems and Networks (DSN)*, 2010 IEEE/IFIP International Conference on (pp. 211–220). IEEE.

AUTHORS

Jim Jones (jjonesu@gmu.edu) is an associate professor at George Mason University. Jones' research activities are focused on digital forensics, specifically the extraction, processing, analysis, and manipulation of digital artifacts. He has been a cybersecurity and digital forensics practitioner and researcher in industry, government, and academia for 20 years. His current and past research has been sponsored by DARPA, DHS, NSF, NPS, DoD, and SAIC. Jones earned a bachelor's in industrial and systems engineering from Georgia Tech, a master's in mathematical sciences from Clemson University, and PhD in computational sciences and informatics from George Mason University.

Carl Beisel (beiselc@gmail.com) is a consultant with Science Applications International Corporation. Beisel is a software and systems engineer with more than 30 years of experience supporting a wide range of government agencies in software development, systems architecture, and project management roles.

A New Five-Factor Process for Increasing Cybersecurity and Privacy

Alireza Aghamohammadi, PhD | Ali Eydgahi, PhD

ABSTRACT

Cybersecurity emphasizes on protecting computers, networks, and applications to allow availability of services and data to authorized users while preventing access to unauthorized individuals or groups. Understanding current security challenges and building solutions to address cybersecurity weaknesses and vulnerabilities are vital keys to increase security and privacy. Web robots are one of the main components of cyber technologies. Web robots or crawlers are utilized by search engines to index and catalog webpages but web robots are also used for spamming, hacking or price fixing purposes. The identification and prevention of unwanted web robots is often very challenging. So, the main purpose of this paper is to present an effective and innovative method to prevent intrusive or unwanted web robots by introducing a five-identifier evaluation process.

In this study, a new method is proposed to prevent unwanted web robots to access websites. This new method utilizes five identifiers of passkey, time, Internet Protocol address lookup, user agent, and number of visitsfor evaluation process of granting access to web robots. The pretest and posttest results along with logistic regression analysis of treatment group and control group are provided. Four hypotheses and quasi-experiment are utilized to evaluate the effectiveness of the proposed five distinct identifiers process. The proposed process provides more effective way of preventing unwanted web robots.

INTRODUCTION

The Internet certainly has played a very critical role for enabling many people to connect and share information quickly and easily. The ability to connect and share information is a luxury that some take as granted in today's world. The Internet has changed the way people communicate and collaborate and it continues to evolve and grow even to this day. The Internet was a great innovation and a technological advancement when it was first developed, but it was a very small network of computers compared to today's Internet (Nelson & Coleman, 2000). The Internet and the servers hosting websites have rapidly grown (Kogut, 2004). For example, around 1994 only 2.2 million web hosts existed on the Internet, but the number has increased to 94 million in 2000 (Kogut, 2004).

The Internet or linked networks of computers began in 1969 with an experiment using only four computers by U.S. department of defense agency called Advanced Research Projects Agency (ARPA) forenabling collaboration and communication between scientists in case of a nuclear accident or strike (Nelson & Coleman, 2000). The technology and protocol implemented by ARPA was called TCP/IP, which even to this day the Internet uses this protocol to link computers and create networks (Sathyan, 2010). The Internet and TCP/IP protocol were designed in such a way that allows growth for sharing information, but this growth unintentionally became a problem of its own as finding information became more challenging as the Internet grow. The obstacle of finding information is documented in (Ledford, 2007, p. 3) as "difficult" and "timeconsuming" experience. As the result of the Internet

growth, web users had to remember which web pages they visited and also remember what content each site contained so that to be able to go back to it whenever they need the same information or site.

The use of Internet and the process of finding information and remembering sites are not very easy as humans are not very good with remembering and finding huge amount of data when compared to computers. So, web robots were developed to assist the Internet users and search engines to collect information from webpages and process them for building indexes of webpages to be able to solve the problem of finding information easily without spending a lot of time on the Internet. The steps or process of indexing is almost identical to the process of producing indexes for book chapters. Similar to a book's indexes, indexes help to find information much quicker because indexes can be used to go to find a specific chapter and the page numbersfor a chapter instead of going through every page in a book to find a specific chapter. In comparison, web indexes help search engines find specific information and a location for a webpage on the Internet. The first web robot software with indexing functionality was developed by Matthew Gray in 1993 (Kuusisto, 2012).

The Internet provides a platform to create and share information and web robots provide a mechanism to find information systematically. The creating and sharing information on the Internet have even evolved from early days of the Internet. Currently, many people use social networks to create information and share pictures and images of themselves on the Internet. However, creating and sharing information on the Internet especially through social network sites increases privacy treats and security vulnerabilities (Albesher, & Alhussain, 2013). Figure 1 depicts software security threats for social networking in terms of cybersecurity and privacy weakness. This paper focuses on web scraping and more specifically the web robots as it pertains to increased cybersecurity and privacy instead of covering all aspects of the Internet weakness and vulnerabilities. Covering all aspects of the Internet weaknesses and vulnerabilities is a broad topic and it cannot be included in one study.

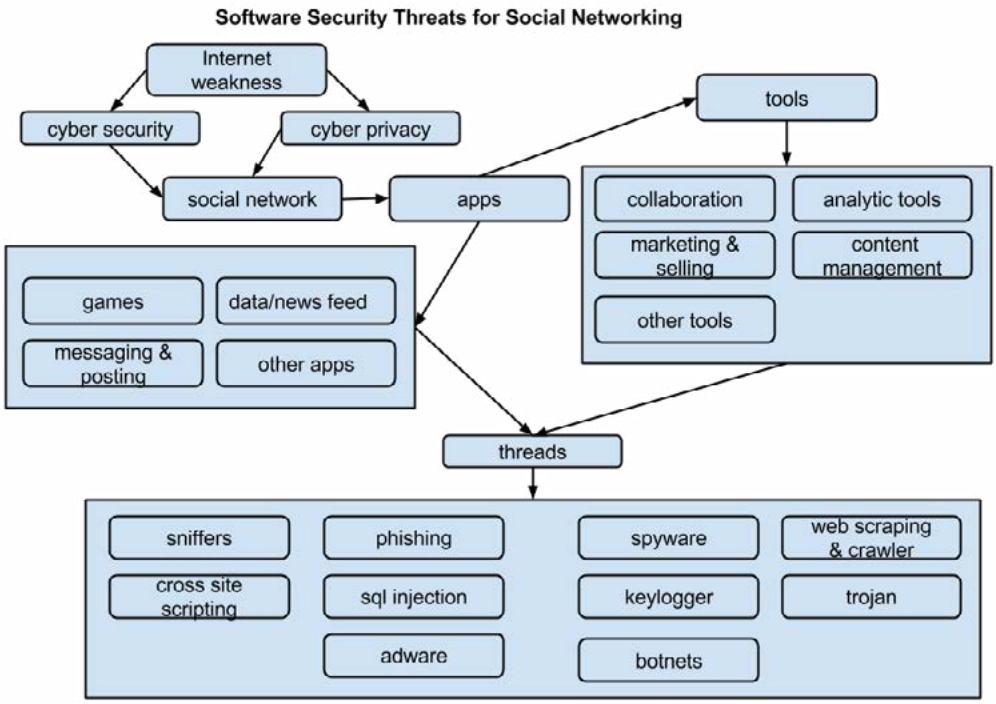

FIGURE 1. SOFTWARE SECURITY THREATS FOR SOCIAL NETWORKING

Web scraping is the "process of collecting unstructured or semi-structured information from the World Wide Web at different levels of automation" (Kokkoras, Ntonas, & Bassiliades, 2013, p. 9). Web scarping involves using a web robot to collect data from various websites. The search engines have improved and evolved in terms of collecting methods, processing algorithms and technology compared to Mathew Gray's basic web robot. However, the improvements and advancements for faster and better web robots have not always benefited web users, government entities and business communities because web robots are usually utilized with one of the two followings goals.

First, web robots or scrapers are used by search engine organizations to index websites to make the websites searchable and more easily available for web users. Second, web robots or scrapers are sometimes used for a more unethical and even criminal act because in some cases web robots are used to collect information or penetrate through a website for stealing information (Sun, 2008). As shown in Figure 1, using web robots to collect information is a cybersecurity treat and invasion of privacy. Multiple studies have reflected this misuse of web robots and the need to better distinguish web robots in order to block and keep away unwanted web robots (Stassopoulou, & Dikaiakos, 2009; Doran, & Gokhale, 2011).

This paper investigates and examines a new approach to identify and restrict unwanted web robots access by using a Five-factor identification process while allowing valid or wanted web robots to still access webpages and information to increase cybersecurity and reduce invasion of privacy. However, measuring the overall security and privacy of webpages were outside of scope of this study rather measuring success or failure of web robots to access webpages were measured.

RELATED WORKS

Web robots have been studied by other researchers previously. However, this research is different from previous works and researches because previous studies focused on different topics, approach and solutions related to web robot. Previous studies can be categories into the following topics:

Web caching and performance optimization

One of the topics previous studies examined is about performance and caching algorithms of web robots (Giles, Sun, & Councill, 2010; Douglis, Feldmann, Krishnamurthy, & Mogul, 1997; Krishnamurthy, Mogul, & Kristol, 1999). The main purposes of these studies were to create new solution for performance improvements and algorithms of web crawling or robots.

Ethical issues related to independent web agents and web robots

While some studies focused on performance improvements and algorithms other pervious researchers examined the ethical issues of autonomous agents such as web robots or crawlers (Eichmann, 1995; Sun, 2008; Dittrich, Bailey, & Dietrich, 2009; Gangadharan & Pretorius, 2010). The ethical issues pertaining to web robot is a very complex topic because the guidelines are sometimes very difficult in terms of concluding what is legal or what is ethical.

Web robot detection and Cloaking

Among previous studies, the most related topic pertaining to this research is about how to identify web robots. This topic was covered in two main dimensions.First was about the misuse of web robots by hackers, etc. Second dimension which previous researchers have examined was about methods and approaches to distinguish and keep away unwanted web robot or robots from accessing webpages. Some of the studies focused on Robots Exclusion Protocol, Meta Tags or X-Robots-Tag and how each can be implemented to better protect information on websites (Kolay, D'Alberto, Dasdan, & Bhattacharjee, 2008; Sun, Zhuang, & Giles, 2007). Also, some studies, examined some of the weaknesses of using Internet Protocol address only to distinguish different web robots and how that may not be very reliable (Tan, & Kumar, 2002). Some studies even attempted to characterize web

robots behavior by examining logs (Dikaiakos, Stassopoulou, & Papageorgiou, 2003). One of the solutions which is even still in use by various websites is called Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA). This method is proposed using distorted image as a test mechanism to distinguish humans versus web robots because it is relatively easy for humans to process a distorted image of alphanumeric characters by simply typing them into a text box. But, web robots would typically fail to perform this task (Von, Blum, & Langford, 2004).

Recently, more clickstream mechanism was proposed to simply see the pattern and tracking the clicks on links or images by users and robots (Lourenco, & Belo, 2006; Wang, & Lee, 2011). Web robots usually try to navigate or click on all links or images on a webpage versus humans which only select a few links or images on a page to click on. In addition, to the existing challenges of accurately and easily identifying web robots, a new challenge has been documents by the researchers about cloaking (Wang, Savage, & Voelker, 2011; Wu, & Davison, 2006).

Cloaking is a method to display different text, images, sounds, videos and animations to humans and web robots (Lin, 2009). Cloaking is not tolerable anymore for most search engines since cloaking prevents search engines to see and process the actual content of a website as if a human user was viewing a webpage instead of a web robot (Lin, 2009). So, a new solution is needed to prevent cloaking while keeping away unwanted web robots.

Deep Web and Web robots

Most of current web robots are very effective for indexing static websites but indexing dynamic websites are very hard for most web robots. The dynamic websites or pages are very different because they can only be viewed after a query is posted to a server (Artail, & Fawaz, 2008). So, some of the previous studies attempted to address the challenge of collecting and gathering data from deep web and those webpages which dynamically are generated (Ke, Deng, Ng, & Lee, 2006; Ntoulas, Zerfos, & Cho, 2005; Cafarella, M.J., Halevy, A., & Madhavan, J., 2011).

Miscellaneous study related to web robot

Lastly, some of the researches are related to web robots but these studies are very different from the previously documented topics discussed earlier in this section. For instance, some researches are about recreating webpages by using web robots in cases where backup recovery of a webpage fails (McCown, & Nelson, 2006). This study proposes a process of rebuilding and restoring websites from Google, Yahoo and MSN cached information. Another research attempted to address the security by using web robots as a tool to find and recognize malicious software on the Internet (Likarish, & Jung, 2009). Also, there is a research on using web robots for developing a digital library (Pant, G., Tsioutsiouliklis, K., Johnson, J., & Giles, C.L., 2004).

METHODOLOGY

The process presented in this paper is focused on a systematic research method for measuring and analyzing data in order to best generate results, evaluate hypothesis and conclude the outcome of the study. An experimental approach was utilized for this research in terms of methodology and research design since this study examined and analyzed cause and effect relationship of the proposed Five-factor identification process on websites to prevent unwanted web robots while allowing access for valid web robots. As stated in (Leedy, & Ormrod, 2005, p. 217) "a researcher can most convincingly identify cause-and-effect relationships by using experimental design".

Since experimental design has multiple types, this study examined each type of experimental design in order to select the best possible methodology approach. For example, the Pre-experimental designs can be applied for researches that are very difficult to examine the cause and effect relationship "because either (a) the independent variable does not vary or (b) experimental and control groups are not comprised of equivalent or randomly selected

individuals" (Leedy, & Ormrod, 2005, p. 223). Since this research had various control groups and the groups had the same sizes in pre and post-test groups, this method was not selected.

Furthermore, by applying only pre-experimental designs it is more difficult to systematically conclude the cause and effect relationship as researchers may not be able to determine the complete changes in context of variables and data. A better method such as True experimental design or Quasi-experimental can be applied and utilized to reduce or eliminate some of the drawbacks of pre-experimental design such as lack of making sure control groups are similar by evaluating them before or after conducting the study. True experimental approach provides a much greater control and improves outcomes with better internal validity assample population is selected randomly and completely by chance (Leedy, & Ormrod, 2005).

One of the key benefits of using True experimental design is the ability to choose random samples from a population. However, this is not possible for all types of studies and populations. In studies where a True experimental research design is very difficult or impossible to implement, a quasi-experimental could be used as a substitute method to examine a cause and effect relationship (Pew, & Hemel, 2004).

The proposed Five-factor identification process uses passkey, time, Internet Protocol address lookup, user agent, and number of visits (allowed each day) as its identifiers.This study utilizes Nonrandomized Control pretest-posttest group as it completes an experiment to investigate whether the proposed Five-factor process can effectively prevent web robots entering a website or a server by utilizing nonrandom samples. The Nonrandomized Control group pretest-posttest group is best defined as a method between the static group comparison which is one of a pre-experimental design types and pre-test-posttest control group design. Nonrandomized Control group is even recognized for its advantage over randomized Control group for some cases since it includes two groups without random selections and it is very similar to static group comparison although it utilizes pretreatment observation in the same way as pretest-posttest control group of True Experimental design (Leedy, & Ormrod, 2005).

MEASUREMENTS

For this study, dependable and independent variables were measured for pretest and posttest steps. Only the counts of success or failure of web robots for visiting a webpage was the dependent variable measured for this study. This approach was based on previous measurement approaches (Kumar, & Vig, 2009; Radhakishan, Farook, & Selvakumar, 2010). For instance, value of dependent variable is set to one if web robot downloads a webpage because web robot ability to download is a failure of process to prevent and protect webpages. In other cases where a webpage cannot be downloaded by web robots, the value of dependent variable is set to zero to indicate a successful process to prevent web robots. This approach will always set the value of dependent variable to zero or one and helps in measuring success or failure of web robot for visiting a webpage.

In addition to measuring dependent variable the independent variables were measured for this study too. The followings are the list of all variables along with explanation of each variable where the success represented with value zero and fail is represented with value one:

- *s* dependent variables – indicates web robot visits status .
- *u*independent variable - indicates web robot's user agent match.
- *t* independent variable - indicates web robot's time match.
- *p* independent variable - indicates web robot's passkey match.
- *i* independent variable - indicates web robot's Internet Protocol address match.
- *v* independent variable- indicates web robot's Number of Visits match for each day.

RESULTS

In this study, data from 720 webpages are collected. The webpages were visited and downloaded by two types of web robots; a good/wanted web robot and a bad/unwanted web robot.

The results were based on 9 groups with each group containing 90 webpages. The webpages were spread across multiple computes. For each group, 9 webpages were created for crawling on each computer. The study utilized one server for hosting the web robots and 10 computers for simulating a small network of computers with 720 webpages. Two Web robots were used for this study as indicated earlier for pretest and posttest steps as shown in Table 1. The groups were categorized to group one (indicating treatment was not applied) and group two (indicating the Five-factor identification/treatment was applied) only to posttest step. The Table 1 presents information about all the webpagesthat web robots attempted to download by test type, web robot type and group type.

TABLE 1. SAMPLE DEMOGRAPHIC

Web Page Count	Test Type	Web Crawler Type	Group Type
90	pretest	unwanted	group_1
90	posttest	unwanted	group_1
90	pretest	unwanted	group_2
90	posttest	unwanted	group_2
90	pretest	wanted	group_1
90	posttest	wanted	group_1
90	pretest	wanted	group_2
90	posttest	wanted	group_2

Web robots downloaded 623 webpages from total of 720 webpages because 97 webpages were not downloaded. By examining the 97 webpages, the results showed web robots could not download 8 pages as the result of webpages not being accessible over the network. The data pertaining to these 8 webpages were still included as part of this study because similar cases can occur if webpages were on the Internet with many computers attempting to download a webpage.

So, web robots could not visit and download 89 pages from those 97 webpages because the webpages identity did not equal to the expected values of the access keys and as a result of differences between keys download permission were denied. The results indicated a positive outcome for total number of pages downloaded and prevented, because 13.47% of total webpages were prevented to load and access was denied to each vesting web robot. Five-factor identification process did prevent some of the web robots to download but 86.53% of all webpages were loaded and accessed by web robots as shown in Figure 2.

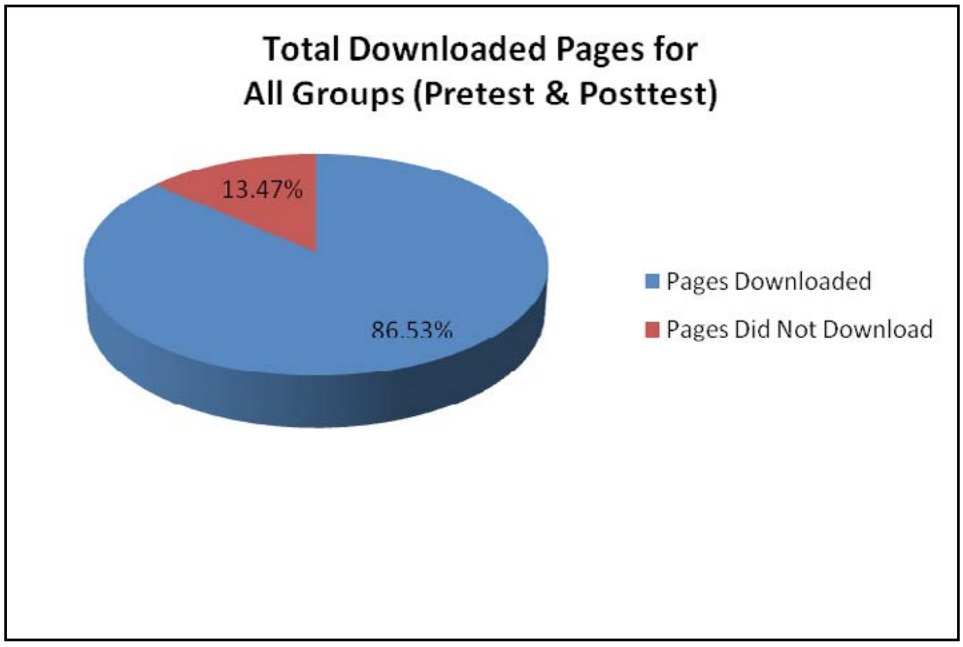

FIGURE 2. DOWNLOADED PAGES

The classification Table 2 and Table 3 were the output of Binary logistic regression from computation by SPSS software. The tables provided in this section show the correctly predicted percentage value for given data sets which are based on observed data that were processed by SPSS software. The classification data in the tables is a valuable indicator to confirm the computed data corresponds to observed data. The classification tables can be most effective when they are read from right to left as these columns provide the most valuable information. In addition, Table 2 and Table 3 reflect the dependent variable variance and web crawling success or failure results. There are two classification tables since two types of web robots are used to test webpages. First table is unwanted web robots classification results and second table is wanted web robots classification results.

TABLE 2. UNWANTED WEB ROBOTS CLASSIFICATION RESULTS

Observed		Predicted		
		Downloaded		Percentage Correct
		Success	Fail	
Downloaded	Success	89	0	100.0
	Fail	1	90	98.9
Overall Percentage				99.4

The classification result in Table 2 depicts the web robots downloads and suggests 89 pages downloaded and 91 pages did not download from total of 180 pages. In addition, the results are based on the analyzing and comparing treatment group and control group for the unwanted web robots only.

The Percentage Correct column in Table 2 and Table 3 is used to depict success rate of analysis by SPSS software for predicting observed data versus actual values. The most useful value for Table 2 is predicted value of the Overall Percentage indicating 99.4%.

TABLE 3. WANTED WEB ROBOTS CLASSIFICATION RESULTS

Observed		Predicted		
		Downloaded		Percentage Correct
		Success	Fail	
Downloaded	Success	178	0	100.0
	Fail	2	0	.0
Overall Percentage				98.9

Table 3 presents outcome of SPSS analysis as it pertains to the number of success and failure of downloads. However, the main difference between Table 2 and Table 3 is the type of web robot utilized for the gathering data including the outcomes differences which are reflected in each table. Success indicates webpages were downloaded successfully by web robot and failed means web robot was denied access to download a webpage. In Table 3, the wanted web robots attempted to download 190 webpages and from total of 190 webpages, only 178 pages successfully downloaded and two did not download. The important number for Table 3 is the Overall Percentage, similar to Table 2. Table 3 values and Table 2 values are different for the most part with one exception for the value of Predicted Percentage Correct for Downloaded Success.

RESEARCH QUESTIONS/ HYPOTHESES RESULTS

In this section, the hypothesis examination and evaluation are presented. The evaluation and examination in this section includes the hypotheses outcome in terms of rejecting or not rejecting each hypothesis. This study constructed two groups prior to completing the pretest and posttest steps and each group contained two hypotheses to better understand the success or failure of Five-factor identification process. The P-values were used to identify any significant effect on the results for treatment and control group. The results were evaluated for unwanted and wanted web robots. Table 4 shows the analysis output perform by SPSS for the P-values calculations:

TABLE 4. P-VALUES FOR TREATMENT/INTERVENTION GROUP AND CONTROL GROUP

Type	P-value	Conclusion
unwanted web robot webpages	0.000	Reject
wanted web robot webpages	0.097	Do not Reject

The hypotheses in group A are (for wanted web robots accessing):

H_0: There is no significant difference between control group and treatment/intervention group, as it pertains to wanted/valid web robots visits.

H_1: There is a significant difference between control group and treatment/intervention group, as it pertains to wanted/valid web robots visits.

Omnibus Test and Binary Logistic Regression were used for this study and the results are shown in Table 4. Omnibus Test, is one of the precise statistically methods to determine if "there is a difference between groups (two or more)" (Swanson, & Holton, 2005, p. 350). P-value of 0.097 was calculated based on Omnibus Test and Binary Logistic Regression. The calculated P-value for Wanted web robots exceeded the .05 alpha level given the 95% confidence interval. The hypothesis test outcome suggests not rejecting H_0 as depicted in Table 4.

The hypotheses in group B are (for unwanted web robots accessing):

H_0: There is no significant difference between control group and treatment/intervention group, as it pertains to unwanted web robots visits.

H_1: There is a significant difference between control group and treatment/intervention group, as it pertains to unwanted/valid web robots visits.

There was a significant change since the p-value was less than 0.05 alpha level given the 95% confidence interval after comparing the results of unwanted web robot for control group and treatment group based on Table 4. The output of a hypothesis testing indicates rejecting H_0 in favor of H_1 as depicted in Table 4.

CONCLUSIONS

This research considered the use of a new method for preventing and restricting unwanted web robots to increase cybersecurity. Various quantitative measurements were used along with binary logistic regression to test the proposed Five-factor identification process. The results from this study suggest there was a significant difference between control group and treatment/intervention group, when Five-factor identification mechanism was introduced to groups. However, there was no significant difference between wanted web robots and their ability to download webpages. This confirms utilizing the proposed Five-factor identification process contributes to the process of preventing unwanted web robots and increases security and privacy. Also, the results indicate use of Five-factor identification continues to allow wanted web robots to access webpages for cases where search engine robots may still need to access the pages.

The analysis and outcomes of this research provide useful and important information for current web robots prevention in cybersecurity and cyber privacy fields.The use of this successful new Five-factor identification process prevents and restricts unwanted web robots intrusion.

REFERENCES CITED

Albesher, A., &Alhussain, T. (2013). Privacy and Security Issues in Social Networks: An Evaluation of Facebook. In *Proceedings of the 2013 International Conference on Information Systems and Design of Communication* (pp. 7–10). New York, NY, USA: ACM. doi:10.1145/2503859.2503861

Artail, H., &Fawaz, K. (2008). A fast HTML webpage change detection approach based on hashing and reducing the number of similarity computations. *Data & Knowledge Engineering, 66*(2), 326 – 337. doi:10.1016/j.datak.2008.04.003

Cafarella, M. J., Halevy, A., & Madhavan, J. (2011).Structured data on the web. *Commun. ACM, 54*(2), 72–79. doi:10.1145/1897816.1897839

Dikaiakos, M., Stassopoulou, A., & Papageorgiou, L. (2003).Characterizing Crawler Behavior from Web Server Access Logs.In *E-Commerce and Web Technologies* (Vol. 2738, pp. 369–378). Springer Berlin Heidelberg. Retrieved from http://dx.doi.org/10.1007/978-3-540-45229-4_36

Dittrich, D., Bailey, M., & Dietrich, S. (2009). *Towards Community Standards for Ethical Behavior in Computer Security Research*. Retrieved from http://staff.washington.edu/dittrich/papers/dbd2009tr1-20090925-1133.pdf

Doran, D., & Gokhale, S. S. (2011). Web robot detection techniques: overview and limitations. *Data Mining and Knowledge Discovery, 22*(1-2), 183–210.

Douglis, F., Feldmann, A., Krishnamurthy, B., & Mogul, J. (1997). Rate of change and other metrics: a live study of the world wide web. In *Proceedings of the USENIX Symposium on Internet Technologies and Systems on USENIX Symposium on Internet Technologies and Systems* (pp. 14–14). Berkeley, CA, USA: USENIX Association. Retrieved from http://dl.acm.org/citation.cfm?id=1267279.1267293

Eichmann, D. (1995). Ethical Web agents. *Computer Networks and ISDN Systems, 28,* 127–136. doi:10.1016/0169-7552(95)00107-3

Gangadharan, V. P., & Pretorius, L. (2010). Towards an ethical analysis of the W3C Web services architecture model. In *Information Security for South Africa*. doi:10.1109/ISSA.2010.5588642

Giles, C. L., Sun, Y., &Councill, I. G. (2010). Measuring the web crawler ethics. In *Proceedings of the 19th international conference on World Wide Web* (pp. 1101–1102). New York, NY, USA: ACM. doi:10.1145/1772690.1772824

Ke, Y., Deng, L., Ng, W., & Lee, D.-L. (2006). Web dynamics and their ramifications for the development of Web search engines. *Computer Networks, 50*(10), 1430 – 1447. doi:10.1016/j.comnet.2005.10.012

Kogut, B. M. (2004). *The Global Internet Economy*.Mit Press. Retrieved from http://books.google.com/books?id=KS3IPQbeINcC

Kokkoras, F., Ntonas, K., & Bassiliades, N. (2013). DEiXTo: A Web Data Extraction Suite. In *Proceedings of the 6th Balkan Conference in Informatics* (pp. 9–12). New York, NY, USA: ACM. doi:10.1145/2490257.2490297

Kolay, S., Dalberto, P., Dasdan, A., &Bhattacharjee, A. (2008). A larger scale study of robots.txt. In *Proceedings of the 17th international conference on World Wide Web* (pp. 1171–1172). New York, NY, USA: ACM. doi:10.1145/1367497.1367711

Krishnamurthy, B., Mogul, J. C., &Kristol, D. M. (1999).Key differences between HTTP/1.0 and HTTP/1.1.*Computer Networks, 31*(11-16), 1737 – 1751. doi:10.1016/S1389-1286(99)00008-0

Kumar, M., &Vig, R. (2009). Design of CORE: context ontology rule enhanced focused web crawler. In *Proceedings of the International Conference on Advances in Computing, Communication and Control* (pp. 494–497). New York, NY, USA: ACM. doi:10.1145/1523103.1523201

Kuusisto, F. (2012). XRDS: Crossroads, The ACM Magazine for Students - The Role of Academia in the Startup World. *XRDS, 18*(4), 41. doi:10.1145/2173637.2173654

Ledford, J. L. (2009). *SEO Search Engine Optimization Bible*.Wiley. Retrieved from http://books.google.com/books?id=mXKiT59e0EEC

Leedy, P. D., & Ormrod, J. E. (2005). *Practical Research: Planning and Design* (8th ed.). Upper Saddle River, NJ: Prentice Hall.

Likarish, P., & Jung, E. (2009). A targeted web crawling for building malicious javascript collection. In *Proceedings of the ACM first international workshop on Data-intensive software management and mining* (pp. 23–26). New York, NY, USA: ACM. doi:10.1145/1651309.1651317

Lourenco, A. G., & Belo, O. O. (2006). Catching web crawlers in the act. In *Proceedings of the 6th international conference on Web engineering* (pp. 265–272). New York, NY, USA: ACM. doi:10.1145/1145581.1145634

McCown, F., & Nelson, M. L. (2006). Evaluation of crawling policies for a web-repository crawler. In *Proceedings of the seventeenth conference on Hypertext and hypermedia* (pp. 157–168). New York, NY, USA: ACM. doi:10.1145/1149941.1149972

Ntoulas, A., Zerfos, P., & Cho, J. (2005). Downloading textual hidden web content through keyword queries. In *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries* (pp. 100–109). New York, NY, USA: ACM. doi:10.1145/1065385.1065407

Pant, G., Tsioutsiouliklis, K., Johnson, J., & Giles, C. L. (2004). Panorama: extending digital libraries with topical crawlers. In *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries* (pp. 142–150). New York, NY, USA: ACM. doi:10.1145/996350.996384

Pew, R. W., & Van Hemel, S. B. (2004). *Technology for Adaptive Aging.* National Academies Press. Retrieved from http://ezproxy.emich.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=109204&site=ehost-live&scope=site

Radhakishan, V., Farook, Y., & Selvakumar, S. (2010). CRAYSE: design and implementation of efficient text search algorithm in a web crawler. *SIGSOFT Softw. Eng. Notes, 35*(4), 1–8. doi:10.1145/1811226.1811236

Sathyan, J. (2010). *Fundamentals of EMS, Nms and OSS/BSS*.Taylor & Francis. Retrieved from http://books.google.com/books?id=7vv1PQAACAAJ

Stassopoulou, A., & Dikaiakos, M. D. (2009). Web robot detection: A probabilistic reasoning approach. *Computer Networks, 53*(3), 265 – 278. doi:10.1016/j.comnet.2008.09.021

Sun, Y. (2008). A comprehensive study of the regulation and behavior of web crawlers (p. 104). University Park, PA: Pennsylvania State University. Retrieved from http://ezproxy.emich.edu/login?url=http://search.proquest.com/docview/231557647?accountid=10650

Sun, Y., Zhuang, Z., & Giles, C. L. (2007). A large-scale study of robots.txt. In *Proceedings of the 16th international conference on World Wide Web* (pp. 1123–1124). New York, NY, USA: ACM. doi:10.1145/1242572.1242726

Swanson, R., & Holton, E. (2005). *Research In Organizations: Foundations And Methods Of Inquiry.* Berrett-Koehler Publishers, Incorporated. Retrieved from http://library.books24x7.com.ezproxy.emich.edu/assetviewer.aspx?bookid=11859&chunkid=376546207

Tan, P.-N., & Kumar, V. (2002). Discovery of Web Robot Sessions Based on their Navigational Patterns. *Data Mining and Knowledge Discovery, 6*(1), 9–35.

Wang, D. Y., Savage, S., & Voelker, G. M. (2011). Cloak and dagger: dynamics of web search cloaking. In *Proceedings of the 18th ACM conference on Computer and communications security* (pp. 477–490). New York, NY, USA: ACM. doi:10.1145/2046707.2046763

Wang, Y.-T., & Lee, A. J. T. (2011). Mining Web navigation patterns with a path traversal graph. *Expert Systems with Applications, 38*(6), 7112–7122. doi:10.1016/j.eswa.2010.12.058

Wu, B., & Davison, B. D. (2006). Detecting semantic cloaking on the web. In *Proceedings of the 15th international conference on World Wide Web* (pp. 819–828). New York, NY, USA: ACM. doi:10.1145/1135777.1135901

AUTHORS

Alireza Aghamohammadi (aaghamoh@emich.edu) is a software engineer at Wellness & Prevention (a Johnson & Johnson company) working on Web services and applications. Prior to joining Wellness & Prevention, Aghamohammadi served as a software engineer at Truven Health Analytics (formerly known as Thomson Reuters healthcare division), where he worked on Web-based internet applications, system integration, and automation. He earned a PhD in 2013 from Eastern Michigan University, and his dissertation focused on Web crawlers detection and prevention. He also holds a BS in computer & information systems from University of Detroit Mercy and an MBA from Wayne State University.

Ali Eydgahi (aeydgahi@emich.edu) joined the College of Technology at the Eastern Michigan University as associate dean in August 2010 and currently is a professor, founder, and director of the Robotics and Autonomous Vehicle Technology Lab in the School of Engineering Technology. He has extensive research work experience and collaboration with different NASA and NAVY centers and has served as a member of review panels for the Department of Education and NASA, as a regional and chapter chairman of IEEE and ASEE, and as a session chair and a member of scientific and international committees for many international conferences. He has published more than 120 papers in refereed international and national journals and conference proceedings.

Towards a Cyber War Taboo? A Framework to Explain the Emergence of Norms for the Use of Force in Cyberspace

Brian M. Mazanec

ABSTRACT

The global community is increasingly dependent on cyberspace. However, as highlighted in the *International Strategy for Cyberspace,* the unprecedented growth of the Internet and growing global reliance on information technology has "not been matched by clearly agreed-upon norms for acceptable state behavior in cyberspace" (United States, 2011). This paper seeks to offer a framework to help explain how norms for cyber warfare are likely to develop. To do so, this paper argues that the scholarly literature on norm evolution as well as case studies on the emergence and development of constraining norms regarding chemical and biological weapons, strategic bombing, and nuclear weapons should be examined. These weapon types each share various similarities with cyber warfare (such as technology with both peaceful and military applications, heightened potential for major collateral damage or unintended consequences, and wide availability of the technology), which make aspects of their norm development experience applicable to the future evolution of constraining cyber norms. This effort fills a gap in the literature by identifying a framework and research agenda to help predict and shape the evolution of norms for cyber warfare, which offer one avenue to contain this growing threat.

"One resists the invasion of armies; one does not resist the invasion of ideas."

-Sir Victor Hugo

INTRODUCTION

In March 2013, James Clapper, the Director of National Intelligence, testified that in just the next two years there is a very real threat that a major cyber attack against the United States would occur, resulting in "long-term, wide-scale disruption of services, such as a regional power outage" (2013). He further stated that the growing international use of cyber weapons to achieve strategic objectives was outpacing the development of a shared understanding or norms of behavior and thus increasing the prospects for miscalculations and escalation (Clapper, 2013). Early in the age of nuclear weapons, Lt. General James Gavin expressed the contemporary wisdom when he said "nuclear weapons will become conventional for several reasons, among them cost, effectiveness against enemy weapons, and ease of handling" (Gavin, 1958). However, as the nuclear era advanced, a constraining norm developed that made states more reluctant to possess or use nuclear weapons—thus helping prevent their widespread diffusion and use. Views similar to those held by Gavin and others at the dawn of the nuclear era regarding military utility and inevitable employment also existed with strategic bombing at the advent of the ability to conduct aerial bombings of civilians during wartime in the early 1900s (Ward, 2001). Today, early into the age of cyber warfare, many hold a similar view regarding the inevitability

of significant use of force in cyberspace. International security and U.S. national security may be enhanced by the emergence of some kind of constraining norm for cyber warfare, similar to those that developed in the past for other emerging-technology weapons. As evidenced by Director Clapper's testimony above, cyber warfare poses a very real threat to U.S. national security. In response to this threat, in May 2011, the Obama administration issued the *International Strategy for Cyberspace* (United States, 2011). One pillar of this strategy recognizes the "borderless" international dimension of cyberspace and identifies the need to achieve stability and address cyber threats through the development of international norms. In February 2013, Michael Daniel, the White House Cybersecurity Coordinator, told computer security practitioners that diplomacy—including fostering international norms and shared expectations—would be essential to preventing cyber warfare against U.S. economic interests, and in March 2013, Tom Donilon, the National Security Advisor, called for China to agree to "acceptable norms of behavior in cyberspace" (Chabrow, 2013; Landler, 2013). This paper seeks to introduce the threat of cyber warfare and the current state of cyber norms. It then suggest a framework and research agenda that may be helpful in better understanding how norms for cyber warfare are developing and will develop in the future. The prospects for the development of cyber warfare norms can best advance through an examination of case studies on the emergence of norms that are in some respects similar, specifically norms for chemical and biological warfare, strategic bombing, and nuclear weapons. While other historical examples regarding norm development may be helpful (such as norms for covert action, assassination, or dueling), these three offer the most promise in developing an understanding of the best way to forge effective cyber norms.

OVERVIEW OF CYBER WARFARE AND INTERNATIONAL NORMS

While there is not a consensus on key terms and definitions regarding cyberspace, the cyber domain is defined by the U.S. Department of Defense (DOD) as the global realm within the "information environment" consisting of the interdependent network of information technology infrastructures, including the Internet, telecommunications networks, computer systems, and embedded processors and controllers (United States, 2011). Some argue that the full electromagnetic spectrum should also be included in any definition of cyberspace, which would make electronic warfare such as radar jamming a form of cyber attack. However such a definition is extremely broad and most have a more limited view. Cyberspace operations are the employment of cyber capabilities where the primary purpose is to achieve objectives in or through cyberspace (Murphy, 2010). Recent examples of cyber conflict were seen in Estonia in 2007, Georgia in 2008, and Iran in 2010 (Healey, 2013). One subset of cyber conflict is cyber warfare—another term lacking a universally agreed-upon definition. On the more violent and serious end of the spectrum, cyber warfare can be described as Computer Network Attack (CNA), which is the use of computer networks to disrupt, deny, degrade, or destroy either the information resident in enemy computers and computer networks, or the computers and networks themselves. This understanding of cyber warfare, conducted between state actors (directly or through plausibly-deniable non-state clients), will be the focus of this paper rather than more-frequent Computer Network Exploitation (CNE), which uses computer networks to gather intelligence on an adversary (United States Government Accountability Office, 2011). As with other forms of warfare, cyber warfare targeting can be counter-value (focused on civilian targets) or counter-force (focused on military personnel, forces, and facilities). Additionally, cyber warfare involves many special characteristics, including the challenges of actor attribution,the multi-use nature of the associated technologies, target and weapon unpredictability,the potential for major collateral damage or unintended consequences due to cyberspace's "borderless" domain, questionable deterrence value, the use of covert programs for development, attractiveness to weaker powers and non-state actors as an asymmetric weapon,and its use as a force multiplier for conventional military operations (Koblentz&Mazanec, 2013). As will be discussed below, many of these characteristics are shared with chemical and biological weapons, strategic bombing, and nuclear weapons, making these three weapon types ideal case studies to develop a

framework for examining the future of cyber norms. Further, many of these characteristics along with the general lack of consensus on what constitutes cyber warfare and even cyberspace itself, highlight some of the challenges facing the emergence of constraining norms for cyber warfare.

Norms are standards of right and wrong that form a prescription or proscription for behavior (Katzenstein, Wendt, & Jepperson, 1996). Essentially, norms are non-binding shared expectations that can be helpful in constraining and regulating behavior of international actors and, in that sense, have a structural impact on the international system. International norms cover a wide range of issues, from norms against the practice of dueling to norms regarding human rights. Specific to warfare, multiple regulative norms have emerged regarding specific categories of weapons and modes of warfare, such as Weapons of Mass Destruction (WMD), strategic bombing, anti-personnel landmines, leadership assassination, and dueling. Norms for weapons and conflict can focus onweapon possession/ development, use, or both. While not always successful (with the demise of the constraining strategic bombing norm in World War II being perhaps one of the best examples), some of these norms for warfare have had an effect in restraining the widespread development, proliferation, or use of various weapons. Therefore, constraining international norms appear as an enticing tool to help address the growing threat opposed by cyber warfare.

NORM EVOLUTION THEORY

There is a wide-ranging and interdisciplinary literature that discussed the emergence and development of international norms. Norms have been utilized as a lens for understanding international activity with increasing frequency, due in part to behavioral and microeconomic research lending support to the tangible role of norms (Goertz, 2001; Rublee, 2009; and Finnemore & Sikkink, 1998). Ann Florini introduced an evolutionary analogy based on natural selection toexplain how international norms change over time (1996). Natural selection, introduced by Charles Darwin in his book On the Origin of Species, is the gradual, non-random process by which biological traits thrive or perish in a population (1859). Norms too can thrive or perish. For example, the initially strong norm against strategic bombing eroded and ultimately perished, due variously to: "inadvertent escalation" resulting from strategic bombing's compatibility with the war-fighting culture of each nation's military services during World War II; the increasing desperation of the state actors and their calculation that the moral opprobrium wrought by violating the norm had become secondary to the existential benefit of using such weapons, and because of pivotal technological change, such as the invention of the Norden bombsight, and improved inertial navigation that made strategic bombing more militarily effective (Legro, 1994; Wrage, 2004).Natural selection entails variation in traits, differentiation (or selection) in reproduction, and replication through hereditary genetics (University of California Berkley, 2013). This evolutionary approach to norms contributes significantly to the theory of norm emergence and development by helping explain why particular norms change over time (Florini, 1996). Norms, like genes, are instructional units that influence the behavior of their host organisms. Genes and norms are both transmitted through inheritance: in the case of norms it is either from one state to another (horizontal reproduction), or internally, within a state (vertical reproduction).Vertical norm reproduction refers to a continuation of a norm through leaders in a single state, and norms reproduced in this way rarely change. In contrast, horizontal norm reproduction is diffusion across multiple states in a single generation. It is this type of norm reproduction that is most relevant when considering norms governing weapon technology and warfare as such norms need to be spread across multiple states in order to influence state-to-state conflict. Overall, norm evolution theory identifies three major stages in a norm's potential life-cycle. These three stages are: (1) norm emergence, (2) norm cascade or tipping point, and (3) norm internalization (Finnemore & Sikkink, 1998). Collectively, these life-cycle stages cover the full spectrum of norm evolution, from the nascent emergence of a novel norm to its near total adoption and codification across the globe. However, a norm may never move through all three stages and can reach its terminal development at any of the three stages and possibly even regress and dissipate.

CURRENT STATE OF NORMS FOR CYBER WARFARE

So where do constraining norms for cyber warfare stand today? In this early stage of the cyber era, norms for cyber warfare emerge in part based on the "general and consistent" practice of states just as much as they arise from deliberate efforts and diplomatic dialogue. Current state practice of CNA-style cyber warfare, which ultimately can develop into customary international law based on "the general and consistent practice of states if the practice is followed out of a sense of legal obligation," can offer some hints as to where cyber norms stand today (Brown & Poellet, 2012). James Lewis and the Center for Strategic and International Studies maintain a rolling list of "significant cyber incidents" since 2006 and identify 153 hostile cyber operations as of July 2013, (2013). While Lewis does not explicitly categorize the operations as either CNE or CNA, the vast majority (137 of 153, approximately 89%) of the incidents appear to be CNE-style operations (Lewis, 2013). That is not to say that there are no CNA-style cyber attacks and that therefore a constraining norm prohibiting such attacks exists. In fact, recently there have been a series of major cyber attacks. Table 1 summarizes these major CNA-style cyber warfare attacks and what they may portend for acceptable norms of behavior in cyberspace, including the suspected sponsor and the target and effect of the attack (Healey, 2013; Singer & Friedman, 2014).

TABLE 1: SELECTED CNA-STYLE CYBER ATTACKS

Attack Name	Date	Target	Effect	Suspected Sponsor
Trans-Siberian Gas Pipeline	June 1982	Soviet gas pipeline (civilian target)	Massive explosion	United States
Estonia	April-May 2007	Commercial and governmental web services (civilian target)	Major denial of service	Russia
Syrian Air Defense System as part of Operation Orchard	September 2007	Military air defense system (military target)	Degradation of air defense capabilities allowing kinetic strike	Israel
Georgia	July 2008	Commercial and governmental web services (civilian target)	Major denial of service	Russia
Conficker	November 2008	Commercial and personal computers (for botnet) and commercial and governmental websites (civilian targets)	Major denial of service	Ukraine
Stuxnet	Late 2009-2010, possibly as early as 2007	Iranian centrifuges (military target)	Physical destruction of Iranian centrifuges	United States
Saudi-Aramco	August 2012	State-owned commercial enterprise (civilian target)	Large-scale destruction of data and attempted physical disruption of oil production	Iran
Operation Ababil	September 2012-March 2013	Large financial institutions (civilian target)	Major denial of service	Iran

The eight CNA-style attacks identified above collectively provide some insight into the emergence of international norms through the customary practice of cyber warfare. There are three main takeaways from the attacks. First, the majority (six of eight) of the attacks were aimed at civilian targets, showing that a norm constraining targeting to explicitly military targets or objectives has not yet arisen. Second, to the extent attacks did strike exclusively military targets, they were suspected to have been launched by Western nations (the United States and Israel). This seems to indicate that there may be competing norms regarding cyber warfare depending on the nation's bloc association—which is consistent with the expected competitive environment in the early days of norm emergence outlined by norm evolution theory. Third, experience with cyber warfare is very limited at this point. No known deaths or casualties have yet resulted from cyber attacks, and the physical damage caused, while impacting strategically significant items such as Iranian centrifuges or Soviet gas pipelines, has not been particularly widespread or severe. While the current absence of massively disruptive cyber attacks is likely due to the limited capabilities and not a constraining norm, the lack of such attacks may allow space for a constraining norm to emerge.

The question of where constraining norms for cyber warfare go from this relatively blank slate can best be addressed through an application of norm evolution theory tailored specifically from some emerging technology weapons that are in some respects similar to cyber weapons. Predictions based on a historical examination of norm evolution in these similar instances could prove insightful and help inform policymakers as they seek to pursue international norms to help manage the cyber threat. As mentioned earlier, constraining norms have emerged regarding other forms of weapons, such as chemical and biological weapons, strategic bombing, and nuclear weapons. While not always successful, some of these norms have had an effect in restraining the widespread development, proliferation, or use of these weapons. Experience with the emergence (and in some cases collapse) of these constraining norms will be particularly helpful in developing an understanding of the future of norms for cyber warfare due to various commonalities between these weapon-types and cyber warfare.

NORMS FOR CHEMICAL AND BIOLOGICAL WEAPONS

Chemical and biological weapons and cyber weapons are both forms of non-conventional weapons that share many of the same special characteristics, with significant international security implications. They include:challenges of attribution following their use; attractiveness to weaker powers and non-state actors as asymmetric weapons; use as a force multiplier for conventional military operations; questionable deterrence value; target and weapon unpredictability; potential for major collateral damage or unintended consequences due to "borderless" domains; the multi-use nature of the associated technologies,and the frequent use of covert programs to develop such weapons (Koblentz & Mazanec, 2013). Due to these characteristics, both of these weapons are also attractive to non-state actors or those seeking anonymity, resulting in a lack of clarity regarding the identity of the responsible party. Because of these common attributes, lessons regarding norm development in the use of cyber weaponscan be learned from applicable chemical and biological weapons experiences. Some chemical and biological warfare norms are codified in contractual obligations and binding agreements such as treaties, as was the case with biological weapons when the Convention on the Prohibition of the Development, Production, and Stockpiling of Bacteriological (Biological) and Toxin Weapons and on Their Destruction, commonly referred to as the Biological Weapons Convention or BWC, entered into force in 1975. The BWC codified the existing norm against development, production, and stockpiling of biological weapons and declared their use to be "repugnant to the conscience of mankind." The BWC now has 155 States Parties (Koblentz & Mazanec, 2013). Earlier norms against use of these weapons led to the 1925 Geneva Protocol's prohibition on their first use in a conflict (states retained their right to retaliate with such weapons). Other binding agreements codifying these norms exist, including the Chemical Weapons Convention, which prohibits outright all chemical weapons. Examining the factors leading to these successes is helpful in developing a framework to predict how constraining norms for cyber weapons may evolve.

NORMS FOR STRATEGIC BOMBING

Strategic bombing—particularly with the advent of airpower and the early use of airplanes to drop bombs on cities—forced states to grapple with a brand new technology and approach to warfare, which is now the case with cyber warfare. As with chemical and biological weapons, strategic bombing shares some special characteristics with cyber warfare.Strategic bombing made civilian populations highly vulnerable, was difficult to defend against, and used technology which also had peaceful applications (air travel and transport)—all of which can also be said about cyber warfare today. At the end of the nineteenth century, technology had advanced to the point where substantial aerial bombing of civilian and military targets from balloons was conceivable. Such "strategic bombing," particularly of civilian targets, appeared to conflict with the existing norm of noncombatant immunity. At the Hague Peace Conference of 1899, the participants agreed to prohibit the "discharge of explosives or projectiles from balloons" for a period of five years (Ward, 2001).The codification of this emerging-technology weapon norm struggled through various debates before, during, and after World War I. However, by the 1930s there was a consensus that bombing civilians was unacceptable, even drawing an admission from Adolf Hitler in 1935 that a "prohibition on indiscriminate bombing of densely populated regions" was warranted (Overy, 2005). However, this emerging norm collapsed during World War II. It eventually remerged from the ashes of the conflict and developed into an enduring norm today. The effort to constrain strategic bombing through normative influences was mixed and at times completely unsuccessful, which makes it particularly well suited as an exemplar of the limits of norms and how other factors may impede or reverse norm development.

NORMS FOR NUCLEAR WEAPONS

Nuclear weapons, like airpower before it and perhaps cyber weapons today, presented states with a challenge of a completely new war fighting technology. Nuclear weapons and cyber weapons, like the other emerging technology case studies, share many of the same special characteristics with significant international security implications.These include the potential for major collateral damage or unintended consequences (due to fallout, in the case of nuclear weapons)and covert development programs. While early nuclear norms were permissive and did not constrain the United States from deploying nuclear bombs on Hiroshima and Nagasaki, that soon changed. As noted by Thomas Schelling, the rapid emergence of norms against the use of nuclear weapons was so effective in constraining action that President Eisenhower's Secretary of State, John Foster Dulles, when contemplating the use of nuclear weapons in 1953 (less than a decade after the first use of nuclear weapons), said that "somehow or other we must manage to remove the taboo from the use of [nuclear] weapons" (Schelling, 2007). This constraining nuclear norm was eventually internalized and codified in agreements such as the Nuclear Nonproliferation Treaty, which enacted limits on nuclear proliferation and a commitment to eventual disarmament. Examining the successful emergence, cascade, and internalization of the constraining nuclear norm may help point a path for success with prospective cyber norms.

A RESEARCH AGENDA TO HELP PREDICT AND SHAPE THE FUTURE EVOLUTION OF NORMS FOR CYBER WARFARE

Each of the three main historical case studies introduced above—in addition to general norm evolution theory—alludes to various important actors, motives and factors that have helped the various norms emerge, grow, or collapse. Some of these important elements and expectations are summarized in Table 2 below.

TABLE 2: SELECT EXPECTATIONS BASED ON NORM EVOLUTION FOR CHEMICAL AND BIOLOGICAL WEAPONS, STRATEGIC BOMBING, AND NUCLEAR WEAPONS

Norm Emergence
Coherence and grafting with existing norms will play a key role in the early foundation of the norm for the emerging technology weapon.
With undemonstrated emerging-technology weapons, there will be challenges. Specifically: • Differing perspectives as to its future capability, which can impair norm emergence. • Prospect for inadvertent escalation to lack of clarity regarding new technology.
Initial weapon proliferation/adoption will play a role in norm emergence
Norm Cascade
Improvements in technology that address previous challenges in adhering to a constraining norm can rapidly lead to a norm cascade.
Characterizing the weapon-type as "unconventional" or otherwise granting it a special status can accelerate norm adoption and ultimately achievement of a norm cascade.
The international arms control and disarmament bureaucracy and the increasing regulation and legalization of armed conflict provide an increased number of organizational platforms and networks to spread the norm and more rapidly achieve a norm cascade.
Norm Internalization
Internalization of aspects of a norm governing usage occurs more rapidly and is easier to achieve than aspects governing development, proliferation, and disarmament.
Secrecy associated with emerging-technology weapon programs and the possible multi-use nature of their technology will impede norm evolution, especially internalization.

Examining the experience with constraining norms for chemical and biological weapons, strategic bombing, and nuclear weapons in order to further develop and understand these factors and their relevance to the emerging area of cyber warfare will be particularly helpful in developing an understanding of how norms for cyber warfare will (or will not) develop. Additionally, norms for cyber warfare may develop more quickly through recognition and adoption of approaches that have been learned through efforts to encourage the evolution of norms for other emerging-technology weapons. This initial analysis of prospective lessons from these case studies indicates that some of these factors and conditions—such as the ability to graft new norms onto existing normative concepts, categorizing or branding weapons as different, or the importance of utilizing existing organizations to foster norm emergence—that were instrumental in fostering or inhibiting norms for these related emerging-technology weapons will be critical for the emergence of constraining cyber norms. Further work in this area is needed to develop plausible scenarios for how cyber warfare norms may develop, as well as suggested lessons for policy makers seeking to encourage this process. With this

knowledge, effective cyber warfare norms can emerge more rapidly as a mechanism to contain this growing threat. Yogi Berra once said, "In theory there is no difference between theory and practice. In practice, there is." These proposed historic case studies will bridge these two worlds and ground norm evolution theory in practice by refining it specifically for emerging-technology weapons. This new analytic framework could then be used to assesshow similarly constraining norms for cyber warfare may develop and whether or not norms offer a viable avenue to contain the growing cyber menace.

REFERENCES CITED

Brown, G., Poellet, K. 2012. The Customary International Law of Cyberspace.*Strategic Studies Quarterly,* Vol. 6(3).

Chabrow, E. (2013, March 1). Using Diplomacy to Stop Cyber Attack. *GovInfoSecurity.com.*

Clapper, J. (2013, March 12). Statement for the Record: Worldwide Threat Assessment of the U.S. Intelligence Community.*Senate Select Committee on Intelligence.*

Darwin, C. (1859). *On the Origin of Species.* Signet Classics. Retrieved from http://darwin-online.org.uk/

Finnemore, M., Sikkink, K. (1998). International norm dynamics and political change.*International Organization.* Vol. 52: 887–917.

Florini, A. (1996) The Evolution of International Norms. *International Studies Quarterly,* Vol. 40(3): 363–389.

Gavin, J. (1958). *War and Peace in the Space Age.* Harper Brothers.

Goertz, G. (2003). *International Norms and Decision making: A Punctuated Equilibrium Model*. Rowman& Littlefield.

Healey, J. (2013). *A Fierce Domain: Conflict in Cyberspace, 1986 to 2012*. Cyber Conflict Studies Association.

Katzenstein, P., Wendt, A., and Jepperson, R. (1996). Norms, Identity, and Culture in National Security in*The Culture of National Security: Norms and Identity in World Politics*. Columbia University Press.

Koblentz, G., Mazanec, B. (2013). Viral Warfare: The Security Implications of Cyber and Biological Weapons.*Comparative Strategy*Vol. 32(5): 418–434.

Landler, M., Sanger, D. (2013, March 11). U.S. Demands China Block Cyberattacks and Agree to Rules.*The New York Times*.

Legro, J. (1994). Military Culture and Inadvertent Escalation in World War II.*International Security*. Vol. 18(4): 108–142.

Lewis, J. (2013, July 11). Significant Cyber Events since 2006. *Center for Strategic and International Studies*. Retrieved from http://csis.org/publication/cyber-events-2006

Murphy, D. (2010). What is Way? The Utility of Cyberspace Operations in the Contemporary Operational Environment.*United States Army War College Center for Strategic Leadership*. Issue Paper Vol. 1–10. Retrieved from http://www.carlisle.army.mil/DIME/documents/War%20is%20War%20Issue%20Paper%20Final2.pdf

Overy, R. (2005). *The Air War: 1939–1945*. Potomac Books.

Rublee, M.R. (2009). *Nonproliferation Norms: Why States Choose Nuclear Restraint*. University of Georgia Press.

Schelling, T. (2007). The Nuclear Taboo.*MIT International Review*. Retrieved from http://web.mit.edu/mitir/2007/spring/taboo.html

Singer, P.W., Friedman, A. (2014). *Cybersecurity and Cyberwar: What Everyone Needs to Know.* Oxford University Press.

Thomas, W. (2001). *The Ethics of Destruction: Norms and Force in International Relations*.Cornell University Press.

United States Department of Defense. (2011, May 15). *Joint Publication 1-02: Department of Defense Dictionary of Military and Associated Terms.*

United States Government Accountability Office. (2011, July 29).*Defense Department Cyber Efforts: Definitions, Focal Point, and Methodology Needed for DOD to Develop Full-Spectrum Cyberspace Budget Estimates*. GAO-11-695R.

United States. (2011, May).*International Strategy for Cyberspace: Prosperity, Security, and Openness in a Networked World*.

University of California Berkley. Understanding Evolution. Retrieved from http://evolution.berkeley.edu/evolibrary/article/evo_25

Wrage, S. (2004). Compliance with Aerial Bombing Norms: A Study of Two Periods, 1939–1945 and 1990–2004. *Annual Convention of the Joint Services Conference on Professional Ethics.*

AUTHOR

Brian M. Mazanec (brianmazanec@gmail.com) is a defense analyst with professional experience supporting a range of government organizations. He has worked for Congress, the Joint Staff, Office of the Secretary of Defense, Defense Threat Reduction Agency, Department of Homeland Security, and the Intelligence Community. Mazanec holds a BA in political science from the University of Richmond and MS in defense and strategic studies from Missouri State University's Department of Defense and Strategic Studies with a thesis on Chinese cyber warfare. He is a doctoral candidate at George Mason University's School of Public and International Affairs with a dissertation focused on the emergence of norms for the use of force in cyberspace.

The Power of Rails and Industry Collaboration in Cyber Education

Gordon W. Romney, PhD | Miles D. Romney | Bhaskar Sinha, PhD
Pradip P. Dey, PhD | Mohammad N. Amin, PhD

ABSTRACT

A recently accredited MS in Cyber Security program (CSIA), and now, NSA/DHS CAE, selected Ruby on Rails (Rails) as its programming language of focus and has grown, in parallel, as Rails became the defacto "lingua franca" of Internet startups. It was selected for CSIA, at the suggestion of an industry collaborator, because it enforces good coding habits, encourages better security practices, is used in cyber tool creation, and its framework facilitates agile development and course delivery. Rails is open-source and runs in more than 210,000 websites including Twitter, Metasploit, Groupon, Living Social, Shopify, and GitHub. Rails is an interpreted language that makes use of agile, scalable development methodologies and RESTful architecture. Rails was designed specifically for the Internet and has had over 1.3 billion downloads. Gartner forecast that the worldwide population of Ruby developers would grow in five years by 400% to over four million by 2013. With this pedigree, Rails fits cyber security curricula perfectly. It serves as an excellent introductory web environment for beginning online students using virtualization and demonstrates the synergy between a web server (Nginx), a database server (MySQL, Oracle) and a browser. Security issues due to industry usage of SQL are readily evaluated using Rails. Progressing in the Master of Science program, pen testing tools are introduced using the recently released Kali Linux. Metasploit, one of the penetration testing tools, was rewritten for this release using Rails.

Key Words: Agile, cyber security, database, Parallels, Ruby on Rails, VMware, Virtual Box, virtualization

INTRODUCTION

"Sharing knowledge creates synergies and promotes development" is the fundamental premise of *Business, Industry and Academia: Networks and Information Sharing* (Sharing, 2013). Information sharing was initiated by Spork Labs, a collaborator for the past decade with G. Romney, one of the authors, to facilitate the implementation of operating system virtualization in engineering laboratory exercises at both a semester-based university as well as National University that uses a one-semester-course-per-month modality (Lanoy & Romney, 2006; Romney, 2009). An industry partner, such as Spork Labs, is motivated by marketing incentives to recognize and be agile in adopting innovative, leading-edge technology that will provide a competitive advantage. The Rails framework just completed its ten-year anniversary and is based on the object-oriented Ruby programming language. M. Romney, an author of this paper, and Managing Partner of Spork Labs, was an early adopter of Ruby on Rails (Rails). He advocated its use in information technology, computer science and cyber security programs and specifically, in virtual hands-on, experiential lab exercises. Five years into the sharing 'collaboratory,' as NSF has defined such a relationship, Spork Labs, as a member of the Cyber Security and Information Assurance (CSIA) industry advisory council at National University (NU) spearheaded the introduction of Rails into the CSIA Master of Science (MS-CSIA) curriculum. The contribution was industry-to-academia, altruistic and with no anticipation of beneficial reciprocation or any contribution from NU. This partnership was established because Spork Labs had a vision regarding the power of Rails in the security industry. Spork

Labs desired to further the use and adoption of Rails technology by academia and students, who, in turn, would take the knowledge gained into industry and the security profession. In academia, its ease of usage on windows, Linux and OSX platforms made Rails an immediate development tool candidate. During the same time frame, the open source availability, coupled with the agile quality of Rails made it the web-wide development programming language of choice at an accelerating rate, worldwide, as it was specifically designed for Internet use.

Rails is an open-source, object-oriented web programming framework that enforces good coding habits (it advocates "convention over configuration"), encourages better security practices, is used in cyber tool creation, and its framework facilitates agile development and course delivery. Rails is open-source and runs in more than 210,000 websites including Twitter, Metasploit, Groupon, Living Social, Shopify, Basecamp, Scribd, Hulu, Yellow Pages, J.P. Morgan, John Deere and GitHub, (Kazanjy, 2013; Fernandez, 2008; Modis, 2013; Mornini, 2011). One major developer, Engine Yard, that has coded in Rails for seven years, stated "Rails is on fire because it is the most productive way to build web applications" (Mornini, 2011). Over eight million projects in 180 countries with a nine-year, 99.99% uptime reliability record are managed by Basecamp, the seminal Rails application (Basecamp, 2013).

Six fringe benefits resulted from the focus on Rails as the CSIA program evolved, namely, the use of 1) agility in both pedagogy and programming development, 2) Rails as a preferred web development language, 3) Rails core security architecture, 4) virtualization as the delivery technology, 5) Rails facility of switching database engines, and 6) Rails as a security software development tool.

Research and student feedback in the MS-CSIA program encouraged the use of agile teaching methods to include virtualization as a vital teaching tool (Romney, 2009; Dey et al., 2009; Sahli & Romney, 2010; Dey et al., 2012; Romney et al., 2013).The curriculum development for both onsite and online instruction of the MS-CSIA program was based on virtualization usage. The program design met both Western Association of Schools and Colleges (WASC) Accreditation standards, and, also, the Committee on National Security Systems (CNSS) 4011 and 4012 certification requirements (CNSS n.d.). In June, 2013, after four years of planning, implementation, operation and graduation of 80 students, NU, jointly with the MS-CSIA program, was designated a Center of Academic Excellence (CAE) by the National Security Agency (NSA) and the Department of Homeland Security (DHS) (NSA.gov n.d.; NUCSIA.nu.edu n.d.). This achievement is a clear demonstration of the value that agile pedagogy and experiential learning through virtualization bring to the curriculum development and instructional design process.

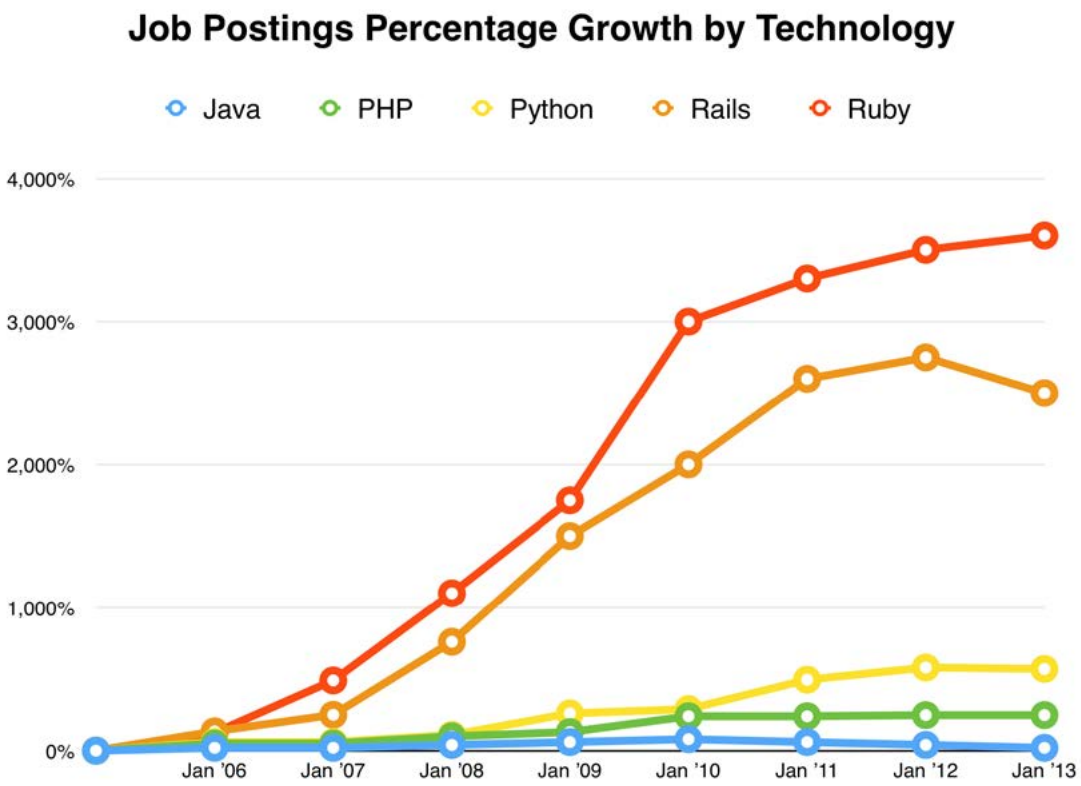

FIGURE 1. PROGRAMMING JOB TRENDS

The authors have been amazed at the exponential growth of adoption of Rails by the information technology, software development and web services industries that have helped produce the above benefits and outcomes in such a short time frame. The foresight of Spork Labs, in 2003, to provide knowledge and assistance to assist the authors to begin using Rails and demonstrate its power in applications it had created is remarkable in retrospect. In 2008, Gartner forecast that the worldwide population of Ruby developers would grow in five years by 400% to over four million by 2013 (Gartner, 2007). Figure 1, Programming Job Trends, compares the percentage growth of Ruby, Rails, Python, PHP and Java through 2011, and reveals both Ruby and Rails significantly outpacing the other contenders (Mornini, 2011).

The Agile Manifesto of 2001

The first benefit derived from the introduction of Rails into the CSIA program was "agility in both pedagogy and programming development." The School of Engineering, Technology and Media (SETM) at National University, for the past seven years, has become an agile "incubator" based upon the Agile Manifesto in software development. Pedagogical agility, agility in student assignments due to the NU one-course-per-month modality, and agility in software development processes have been introduced into SETM curricula by the authors. The emphasis upon "Agility" in engineering and software development was signaled by the Agile Manifesto in 2001 (Agile Manifesto, 2001). Seventeen industry software engineers declared a change in the software development process. One of these, Thomas, became a noted Rails evangelist and publisher, and author of *Agile Web Development with Rails* (Thomas et al.,

2006). Agile software development, unlike the rigid, sequential "waterfall" model for software development, consists of development methods based on incremental and iterative steps. Project requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It facilitates adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages nimble, rapid and flexible response to change. It is a conceptual framework that promotes synergistic interactions throughout the development cycle (Agile, 2013). From a pedagogical or teaching perspective, the flexibility and ease of Rails development for a one-month course project has been consistently demonstrated in NU instruction. Rails and agility are synonymous as Rails is an agile programming tool. Without such agility, course projects of significance would not have been possible. Examples of agility in program development, specifically for one-month course projects, are given later in this paper.

Ruby on Rails

The second benefit derived from the introduction of Rails into the CSIA program was the use of "Rails as a preferred web development language." The MS-CSIA program, by design in order to meet the needs of security managers who desire CNSS 4012 and 4013 certifications, does not have a strong programming prerequisite requirement for admission. CSIA programs do not have course slots to teach programming skills. This is particularly the case in a WASC accredited, CNSS certified twelve month Master of Science program. Consequently, CSIA students are not all strong programmers. Rails, being web-based, security-centric, with Internet presence, superb language architecture and ease of usage, became a ready candidate as an exemplary, secure web programming framework. Rails was designed with web applications in mind and follows the Model-View-Controller (MVC) framework model. An active, very opinionated debate continues between advocates of Python and Rails developers as to which is the more desirable language. Python is procedural-based in contrast to the object-oriented Rails. Scientific users find Python more akin to Fortran and C, more readable, and believe it generally provides unique code solutions. Ruby is a unique open-source programming language that is of growing popularity, as Figure 1 reveals. Rails is a framework that when used with Ruby makes an easy and productive web development framework. This paper covers the ten-year period from the introduction of Rails, as covered in the history of Rails releases (Rails Releases, 2013). The current release version is Rails 4.1.0.

Principles Behind Ruby and Ruby on Rails

Ruby was written by Matsumoto, and first released to the public in 1996 as Ruby 1.0. The popularity of Ruby has only grown since, with many improvements and the creation of the Ruby on Rails framework by Hannson in 2003. Ruby is unique because the basic principles on which it was based, conciseness, consistency and flexibility, were inspired by Perl with Smalltalk-like features (Matsumoto, 2000). These three principles make programming in Ruby not only fun but, more importantly, productive. This is where Ruby stands out to be the language of choice in agile problem-driven teaching environments. The principle of conciseness dictates that a language should do a lot of work quickly. The principle of consistency means that a programmer with basic knowledge can learn Ruby very quickly. Last but not least, the principle of flexibility means that Ruby will help express humans, not restrict them. A Ruby programmer can write arbitrary objects that are treated just like the built-in ones. Ruby is purely object-oriented and is an interpreted language. These two characteristics put every feature of the language in perspective according to the principles behind it. A reflective programming language allows an active environment to query, extend, or modify objects at runtime. As a dynamic language, Ruby implements reflection, allowing a programmer to check type, class, and methods of objects at runtime.

Scott (Scott, 2006) affirms that a big portion of costly application bugs come from programming errors in memory management that are caused largely by poor garbage collection. Memory leaks and dangling pointers are common bugs in applications written in languages that require manual memory deallocation. Ruby has an automatic garbage collector (GC) that

relieves a programmer from performing such a task. Just like mostly anything else in Ruby, the garbage collector is an object that can be accessed and managed using the GC module or the ObjectSpace module (Thomas et al, 2004).

Ruby on Rails Security – Vulnerabilities and Countermeasures

The third benefit derived from the introduction of Rails into the CSIA program was "Rails core security architecture." Plug-n-play security is a fallacy, especially when it comes to application development. Web application frameworks are designed to help programmers use best practices in their coding. Some frameworks, additionally, assist in securing the web application and Rails is among the select few that do. Rails has clever helper methods to mitigate against SQL injection attacks. Hansson, the creator of Rails, obviously took to heart the 2003 finding of the Gartner Group "that out of 300 audited sites, 97% are vulnerable to attack" and estimated that 75% of attacks are at the web application layer. Web applications are comparatively easy to attack as they are simple to understand and manipulate, even by the non-professional (Security Guide, 2013; Grossman, 2003). Ten years later the industry continues to be plagued by SQL injection attacks largely due to the use of insecure programming frameworks and poor security coding practices. Education regarding the techniques and best practices, and discipline in coding must be repeatedly taught and practiced.

The SANS Institute joined, in 2008, with the National Security Agency, national and international security agencies, and private industry to identify the Critical Security Controls that focus on "What Works" in mitigating attack vectors. A list of the Top 20 Security Controls was produced. Automation and measurement of the Top 20 Security Controls achieved more than a 94% reduction in "measured" security risk as reported by the U.S. State Department (SANS, 2013). Only ten percent of the respondents confirmed implementing all twenty of the controls. The fourth control deals with "Continuous Vulnerability Assessment and Remediation," something the Ruby on Rails organization has incorporated in its Rails Security Guide (Security Guide, 2013).

Java and Flash have received significant attention from attackers, but attacks on Rails were relatively silent until January 8, 2013, when two vulnerabilities were identified and immediately patched. The vulnerabilities allow attackers to bypass authentication and perform DoS attacks (Weber, 2013). This is a two-edged sword, however, because identifying the vulnerabilities reveals the weaknesses and one of the major implementers of Rails is Rapid7 with its Metasploit penetration testing software (Metasploit, 2013). Regarding these two Rails vulnerabilities, O'Donnell of Sourcefire stated, "It is my opinion that we have to consider that a worm is not the most serious threat we could face. The worst-case situation is that attackers use the vulnerability to silently compromise massive numbers of vulnerable websites, grab everything from the database, and install persistent backdoors in the infrastructure of every organization running the vulnerable code. They could also silently post a client-side exploit that targets people who come to that site, commonly known as a Watering Hole attack. A worm would likely force everyone to fix their infrastructure immediately, while silent exploitation may not be as motivating" (Fisher, 2013). The benefit of having a company like Rapid7 backing up Rails development is that one can be assured that an organization with capacity to fix a bug will do so immediately. The downside is that Rapid7 also will release an upgrade to Metasploit that includes the attack detail and exploit code. This is an example of true openness and transparency.

Academic instruction regarding the cause of such vulnerabilities, countermeasures and coding best practices is part of the CSIA curriculum at NU and all disciplines in the School of Engineering, Technology and Media and its two departments: Computer Science, Information and Media Systems; and Applied Engineering. The use of Rails, likewise, is a hands-on tool that gives first-hand experience in how to mitigate SQL injection attacks. Our experience is that security awareness and best practices must be taught in both undergraduate and graduate courses in Information Technology (IT), Information Technology Management (ITM), Computer Science (CS), Cyber Security (CSIA) and Information Systems disciplines. This paper deals with the ten-year

experience gained in security instruction, first, in IT, at Brigham Young University, and, subsequently, in ITM, CS and CSIA instruction at National University.

Virtualization

The fourth benefit derived from the introduction of Rails into the CSIA program was"virtualization as the delivery technology." Operating system virtualization, although not a requirement to run Rails, has been a great facilitator in the teaching of engineering and security at NU including the deployment of Rails. Several of the authors recently published a paper describing "The Agility, Flexibility and Efficiency of Hypervisors in Engineering Education" (Romney, 2013, October). Only a summary of these technology concepts will be given in this paper. A Hypervisor is the software that makes the Virtualization process possible. A physical host computer may execute software called a Virtual Machine Monitor (VMM) that is also known as a Hypervisor. A VMM host machine has the capability of running multiple operating systems concurrently referred to as guest machines or Virtual Machines (VMs). A VMM may be software, firmware or hardware that creates and runs virtual machines. A highly significant feature of a Hypervisor is its ability to restrict the operation of each VM to a subset of the host memory space. This makes security of each VM possible.

A software Hypervisor can execute under an existing operating system as an application, or it may be installed natively on bare metal as it has its own operating system kernel. Examples of a Hypervisor executing as an application are a) VMware Workstation under the Windows 8 O/S, or b) Parallels under the Mac OSX, or c) Oracle VirtualBox on Windows, Mac OS X or Linux. An example of a Hypervisor installed on bare metal is VMware vSphere ESXi on a Dell physical computer. Three major attributes of Hypervisors are highlighted in this paper: Agility, Flexibility and Efficiency. The authors in 2009 (Romney, 2009) and 2012 (Dey et al., 2012) stressed the role of "Agility" in engineering education and signaled the fact that NU is an "agility incubator." As Gartner's Bittman said, "Agility is probably the top attribute [one] ... get[s] out of virtualization. For example, [one] ... can deploy servers 30 times faster; if it took two months before, it now takes two days." (Bittman, n.d.)

Efficiency was gained by the ability to provide students with a stable, hardware-independent, virtual machine configuration that ran on a Windows computer on the first day of lab. This saved two weeks of time traditionally lost coordinating all students with different laptops and hardware drivers simply to begin the assignment.

Five years later, in 2009,virtualization became even more important at National University with its one-course-per-month modality (Romney& Juneau, 2009; Romney & Juneau, 2010). Each onsite or online session became all the more critical and hands-on assignments were made even more dependent on the use of virtualization. By using virtualization, flexibility was gained as new concepts could be introduced more readily, resulting in the successful completion of a sophisticated month-course project of a parser written in a new computer language and framework, Rails. Virtualization even facilitated student publishing as Sahli, a graduate student, contributed the parser project as a journal paper (Sahli & Romney, 2009).

Cloud Technology

The use of virtualization technology is particularly useful in the teaching of computer science and information technology curricula. Ever since the advent of computer technology education in the 1960s, academic institutions have invested large sums of capital to equip their programs with the computing equipment necessary to support the learning outcomes defined in computer-related curricula. Increasingly, virtualization provides a greatly enhanced service capability at a significantly reduced cost-per-student. SETM of National University has distinguished itself by providing a Virtual Education Lab (VEL) to support experiential cyber security laboratory exercises that use virtualization (Romney & Juneau, 2009; Romney & Juneau, 2010). The advent of Cloud computing, that leverages virtualization, brings with it the possibility of even greater efficiencies (Romney et al., 2008; Gonzales et al, 2012; Gonzales et al.,

2012, July). The VEL, SETM administered, and its successor, the Information Security Lab Environment (ISLE), NU IT administered, are private clouds and are described in other publications (Anderson & Romney, 2013). Rails VMs were deployed on the VEL from 2008–2013, and the ISLE starting in 2013. The ISLE currently supports over 800 virtual servers, routers and appliances. Virtualization is part of the fundamental technology that has made cloud infrastructures possible and facilitated the rapid adoption of cloud concepts.

SETM students have made, by assignment, use of the NU-provided private clouds, Infrastructure as a Service (IaaS) VEL and ISLE, in the use of Rails. Additionally, students have used public cloud providers such as AppFog as a Platform-as-a-Service (PaaS) to support Ruby and Rails course projects (AppFog, 2013). Further research has been done by the authors using public clouds provided by Engine Yard (Engine Yard, 2013) and GitHub (GitHub, 2013) that are particularly useful for Apple Mac OSX platforms.

THE EVOLUTION OF RAILS USAGE BY THE AUTHORS

Industry-academia collaboration, timely implementation of excellent, secure web technology, and supportive collaboration of diverse disciplines all combined to create a successful cyber security initiative at National University. The authors represent the collegial diversity that made use of Rails for Information Technology Management (ITM), Computer Science (CS) and MS-CSIA programs. The academic authors, G. Romney (BS/MS-IT, Lead Faculty for BS-ITM, MS-CS and Lead Faculty for MS-CSIA), Sinha (Lead Faculty for BS-ITM), Dey (Lead faculty for MS-CS), and Amin (MS-CS and Lead Faculty for MS-Wireless Communications) faculty in the National University School of Engineering, Technology and Media (SETM), have been intimately involved in the integration of security concepts, programming and database design into the courses of their respective disciplines and areas of responsibility. Spork Labs, and M. Romney, have actively contributed Rails technology and support continuously over the time-span of this report (2004–2013), keeping the professors current on upgrades, improvements and applications.

The beginning of Rails usage in 2004 in the IT security program at BYU by G. Romney, one of the authors, was at the suggestion of M. Romney, also an author. As shown in Table 1, Systems Utilized, a local installation of Rails on an Apple MacBook Pro was the start. Free hypervisors for student usage were not available which dictated a local installation on the Mac OSX. Installation of Rails, however, required the separate installation of Apache, Mongrel, Ruby, Ruby Gems, Rails, and MySQL. This was a time-consuming challenge for the beginning student.

TABLE 1. SYSTEMS UTILIZED

Yr	Config	Vendor	Product	System	Database	H/W
04	Local	Apple OSX	No Hypervisor	Rails; OSX	MySQL	32 bit
05	Local	Microsoft	Virtual PC	MSoft XP, Instant Rails	MySQL	32 bit
08	Local	VMware	Workstation	MSoft XP, Instant Rails	MySQL	32 bit
09	Local	VMware	Fusion	MSoft XP, Instant Rails	MySQL	32/64 bit OSX
09	VEL	VMware	ESXi	MSoft XP, Instant Rails	MySQL	32/64 bit
11	Local	Oracle	Virtual Box	MSoft XP, Instant Rails	MySQL	32/64 bit; OSX
11	Local	Parallels	Parallels	MSoft XP, Instant Rails	MySQL	32/64 bit OSX
12	Public Cloud	Application Bundle	AppFog, LAMP, Amazon AWS	MSoft XP, Rails, Ruby	MySQL	32/64 bit
13	ISLE	VMware	ESXi	MSoft XP, Instant Rails, Bitnami	MySQL	32/64 bit
13	Local	Apple OSX	Parallels	OSX, Ubuntu Linux, Rails	MySQL	64 bit
13	Local	Apple OSX	No Hypervisor, Amazon AWS	Rails	MySQL	64 bit

In 2005, Instant Rails, a bundled package of all of the web, database server and Ruby/Rails components was available for Windows XP installation. This opened up the opportunity to use Rails in academic instruction, as a majority of students at both BYU and NU (2007 forward) had Windows-based PCs. The manner in which Rails was now deployed in course instruction became a function of hardware and hypervisor availability. In 2008 Bitnami replaced Instant Rails as a bundled Rails package (Bitnami, 2013).

Spork Labs assisted G. Romney, one of the authors, in creating a Rails application that was MySQL-based and was simply a Message Board application that also demonstrated the insertion of SQL script as Ruby code. Over five hundred students have worked with this application, hardened the security build through several XP service-pack releases and learned about the basic web vs. database server structure of a web site. In the process, they learned a) SQL, b), how to backup and restore a system, c) use other database engines (MS SQL, PostgreSQL, Oracle), d) how to use development tools such as Text Editors, and e) how to transfer/receive data from/into a website. For the majority of the students it was the first time they created a website and understood how its components interact.

The subsequent steps in the evolution of the Systems Utilized figure were the following:

1. 2005–2013. Use of free hypervisors such as Microsoft's Virtual PC, 30-day trials of VMware, and Sun/Oracle's Virtual Box. VM of Message Board Instant Rails was used.

2. 2009–2013. NU obtained a VMware site license that made VMware hypervisors available to students and faculty; also a Microsoft site license that made operating system software available. VM of Message Board Instant Rails was used.

3. 2009–2013. Hypervisor availability made the creation of a private cloud a reality. The VEL is basically an IaaS (Infrastructure as a Service) cloud service that evolved from a research project and involved deployment of four different versions, each building on the features, hardware and software of the previous version. As a proof-in-concept research initiative, system administration was provided by faculty and students. The initial network was deployed in class and gradually evolved to include remote access for onsite students. VM of Message Board Instant Rails was used.

4. 2012–2013. Public Clouds on a trial basis became available for both IaaS and PaaS services. Students developed applications using Ruby, Rails and Rails applications.

5. 2013. ISLE, administered by NU IT staff, with over 800 VMs began operation. VM of Message Board Instant Rails is used. Rails represents a small percentage of the usage of ISLE as MS-CSIA courses use many Linux and Windows servers for complex networking topologies used in penetration testing. Kali Linux, however, uses Metasploit, a Rails application for penetration testing.

6. 2013 Local OSX. Rails on Apple OS X is used in future research projects.

INFORMATION TECHNOLOGY AND INFORMATION TECHNOLOGY MANAGEMENT

Information Technology Management and IT BS-ITM470/475

The Instant Rails Virtual Machine with the Message Board application was used in these two CISSP preparatory security courses. The objective was to harden the security build through several XP service-pack releases and learn about the basic web vs. database server structure of a web site. In the process, they learned a) SQL basics, b), how to backup and restore a system, c) use other database engines (MS SQL, d) how to use development tools such as a Text Editor, and e) how to transfer/receive data from/into a website. This program is administered by Sinha, one of the authors, who is most supportive of introducing security concepts into the curriculum.

Database Concepts & Data Model BS-ITM440

The fifth benefit derived from the introduction of Rails into the CSIA program was "the facility of Rails to switch database engines" as reflected in database courses. The Instant Rails Virtual Machine with the Message Board application was used with an emphasis upon SQL and MySQL database interaction. Students, however, were able to switch to MS SQL and MS Access with minimal difficulty. The use of phpMyAdmin that is a GUI database management interface was also useful. The objective was to learn about the basic web vs. database server structure of a web site. In the process, they learned a) SQL basics, b), how to backup and restore a system, c) use other database engines (MS SQL, PostgreSQL and Oracle, d) how to use development tools such as a Text Editor, and e) how to transfer/receive data from/into a website. The ease of creating additional Rails applications facilitated creative course projects. Having a working template of a relational database as a model was most productive.

MS CYBER SECURITY AND INFORMATION ASSURANCE

Cyber Security Technology CYB600

MS-CSIA students used the Instant Rails Virtual Machine with the Message Board application as an introduction to website architecture and server hardening. The objective was to harden the security build through several XP service-pack releases and learn about the basic web vs. database server structure of a web site. In the process, they learned a) SQL basics, b), how to backup and restore a system, c) use other database engines (MS SQL, PostgreSQL, Oracle), d) how to use development tools such as a Text Editor, and e) how to transfer/receive data from/into a website. This use of Rails is normally done through the VEL or ISLE using a VPN, digital certificates and multi-factor authentication.

Threat Mitigation Policy/Audit CYB602

The sixth benefit derived from the introduction of Rails into the CSIA program was appreciating "Rails as a security software development tool." MS-CSIA students used a Kali Linux Virtual Machine with the Rails-based Metasploit penetration testing application as preparation for future penetration testing and red vs. blue team exercises. Having previously been introduced to Rails, the students gained a better appreciation for the security design of Rails by means of a major Rails application, Metasploit (Metasploit, 2013). The use of a Kali Linux VM was done on a student local machine (Kali Linux, 2013). In this instance the hypervisor used was Oracle's VirtualBox. An example of the Armitage GUI interface for Metasploit is shown in Figure 2, and capturing the password by brute force of the target machine in Figure 3.

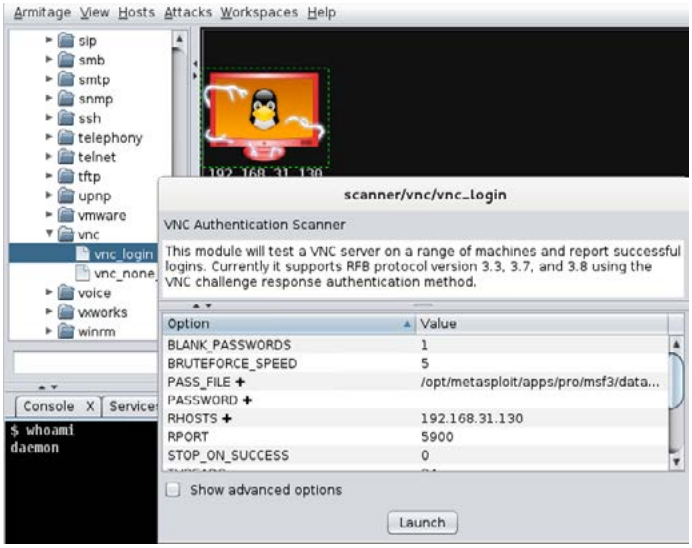

FIGURE 2. ARMITAGE UNDER KALI LINUX

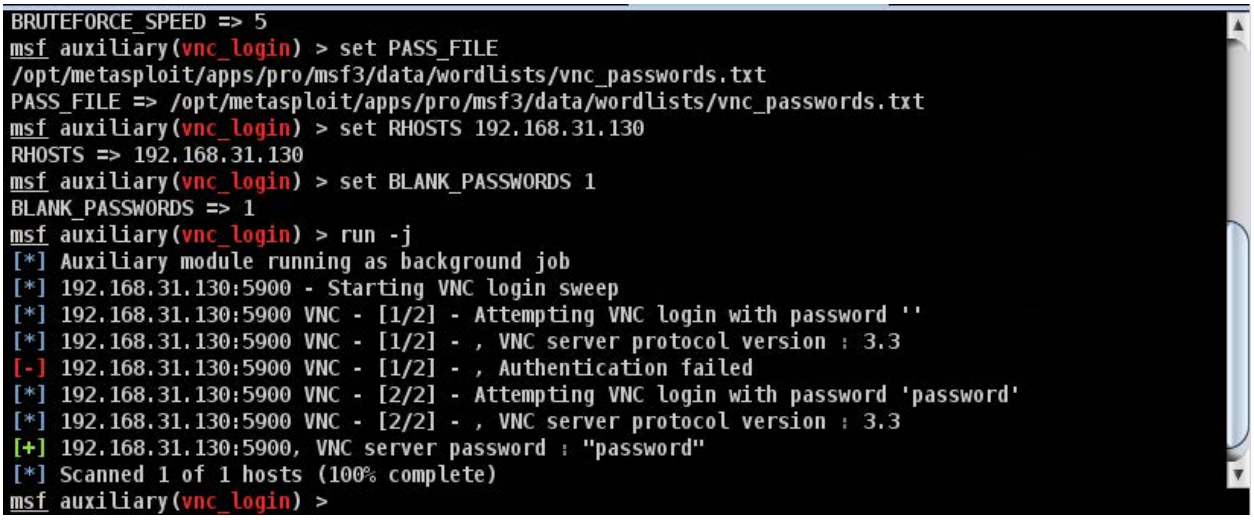

FIGURE 3. PASSWORD OF TARGET MACHINE IS CAPTURED

Armitage is a GUI front end for the Metasploit framework. It allows you quickly and easily to scan, attack, exploit, pivot and attack again. Besides the easy benefits of the GUI, Armitage really excels when it is used in a Red team environment. When used with a team, Armitage shares sessions and vulnerable hosts so that Red Teams can easily share their progress.

MS COMPUTER SCIENCE

Programming Languages CSC650

MS-CSC students used the Instant Rails Virtual Machine with the Message Board application as an introduction to website architecture and Ruby as an object-oriented programming language. These students regularly have C++/C# or Java programming experience and can grasp the strength of Rails architecture and design. One student, a Java programmer by profession, at first emphasized that no language could surpass Java, but accepted the challenge of basing his course project on Ruby. He returned a few days later to the next class and said he was most impressed and would use Ruby to deliver a parser in a month. A journal paper for which he was primary author resulted from this effort and includes a programmer's critical analysis of Ruby (Sahli, 2010). Sahli said the following regarding agility and virtualization: "Last but not least, using virtual machines and other agile teaching techniques for implementing this project allowed the students to learn a great deal about programming languages in only four weeks. Virtual machines are now easy to setup and use. Although we had previously used virtual machines for more traditional purposes, it seems that we missed the more logical use of having them serve as portable and isolated development environments that make agile web development a reality." He also discovered a great tool to package the necessary executable Ruby code into a Windows executable: "The students found a Ruby script that collects and packages all required Ruby files into one Windows executable file, the RubyScript2Exe script (Veenstra, 2007). An executable calculator interpreter was generated and executed in a Windows XP environment."

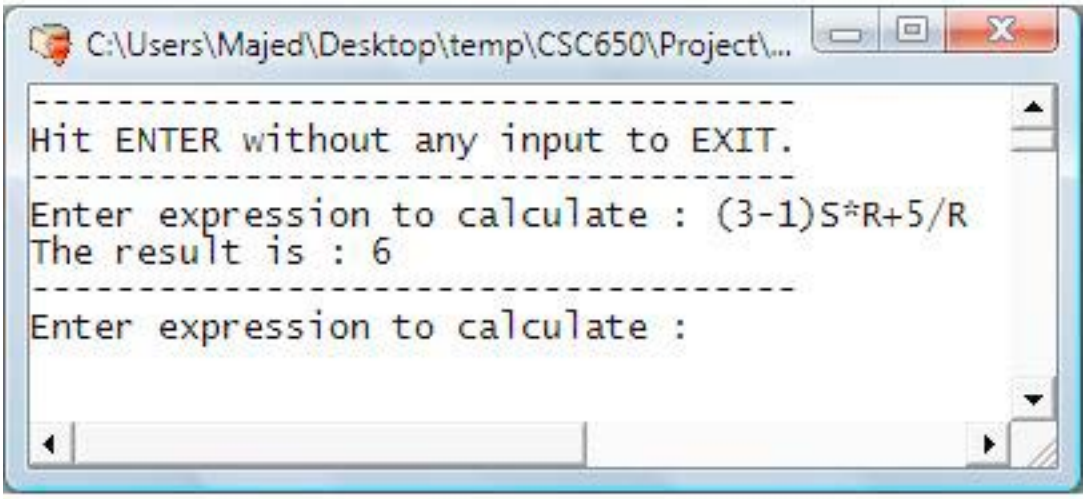

FIGURE 4. A WORKING SOLUTION DELIVERED IN-TIME BY UTILIZING AGILE

Tools in an Accelerated Environment

The grammar logic for parsing the input string in Figure 4 is shown in Figure 5, Grammar Precedence Rules.

(3-1)S*R+5/R,	memory=0	:	Parenthesis has the highest precedence, (3-1)=2.
2S*R+5/R	memory=0	:	The S operator stores the number 2 in memory.
2*R+5/R	memory=2	:	Multiplication precedence is higher than addition
		:	with R evaluated from memory as 2 so 2*R=4.
4+5/R	memory=2	:	Division precedence is higher than addition
		:	with R evaluated from memory as 2 so 5/R=2.
4+2	memory=2	:	Straightforward addition.
6	memory=2	:	Value of the expression.

FIGURE 5. GRAMMAR PRECEDENCE RULES

Web and Cloud Computing DAT605

The fifth benefit derived from the introduction of Rails into the CSIA program was "the facility of Rails to switch database engines" as reflected in database courses. MS-CSC students used the Instant Rails Virtual Machine with the Message Board application as an introduction to website architecture, Ruby as an object-oriented programming

language, Rails as a web framework, and Rails flexibility in using a variety of relational databases. Synergy with the instructor of DAT604, Amin, one of the authors, who most efficiently teaches database normalization, allowed the delivery of a course project that a) emphasizes the flexibility of Rails to create a web application by example, and b) reinforces the concepts of the database design course. Three different options are given those students who have a stronger programming language background, a) use the NU VEL with an Instant Rails Virtual Machine with the Message Board application, or b) use already possessed programming skills either on i) their local machine, or ii) an IaaS or PaaS cloud provider with free service during the course. These students are part of the program administered by Dey, one of the authors, and are consistently creative and deliver surprising projects. Usually this is their first introduction to working in a cloud environment and it brings them great personal satisfaction.

One student used Instant Rails to create a "Tennis Buddy" Social Networking Site for the Cloud and stated, "There are several reasons to use Ruby on Rails to develop web applications. One reason is the speed at which you can develop your project. Even within a short span of four weeks, my social networking site Tennis Buddy has the capability to register users, log them in and out of the site, as well a court listing framework that is capable of being updated by any user. Another reason is the agility of Rails." She created Tennis Buddy on her local machine, as shown in Figure 6, and did research on web hosting and selected the PaaS cloud provider Dreamhost.com as her choice (Dreamhost, 2012).

FIGURE 6. TENNIS BUDDY RAILS APPLICATION FOR THE CLOUD

CONCLUSIONS

Useful Tools Led to Successful Outcomes

Over 120 students have graduated and another 100 are enrolled in both onsite and online cohorts of the National University MS in Cyber Security and Information Assurance program. In June 2013, National University and the MS-CSIA program were honored with designation as a National Security Agency (NSA) and Department of Homeland Security (DHS) Center of Academic Excellence (CAE) in Information Assurance Education (NUCSIA.nu.edu., 2013). The collaboration of industry partners such as Spork Labs that made the significant contribution of vision and knowledge regarding Ruby on Rails described in this paper are the silent partners that deserve credit for the success of the MS-CSIA program.

Six outcomes were achieved from the focus on Rails as the CSIA program evolved, namely, the use of 1) agility in both pedagogy and programming development, 2) Rails as a preferred web development language, 3) Rails core security architecture, 4) virtualization as the delivery technology, 5) Rails facility of switching database engines, and 6) Rails as a security software development tool.

Assumptions and Design Principles for MS-CSIA

- The curriculum was designed to be used first in online and second in onsite instruction.
- A CSIA Advisory Council of industry and academic partners was created in order to incorporate needed skills required to meet the Cyber Warrior demand.
- Ruby on Rails would be used as a preferred web development tool.
- Agile pedagogy and software development methods would be a standard.
- The VEL was created and implemented to provide virtual machines (VMs) for Cyber Security laboratory exercises.
- Course usage of VMs, previous templates, laboratory exercises, the virtual and cloud infrastructure would be designed and specified by SETM faculty and staff, and an industry partner, iNetwork Inc.
- The courses were designed in order to meet the specific security certification requirements of the CNSS standards, 4011 and 4012.
- Both online and onsite instruction and assessment was designed to meet WASC accreditation requirements.
- The objective was to have NU qualify for and be designated as a National Security Agency and Department of Homeland Security Center of Academic Excellence in Information Assurance Education (CAE-IAE).

FUTURE RESEARCH

Future Rails research is already underway in two areas. First, software development versioning and management using two cloud resources, namely, 1) GitHub.com with its code development collaboration tools, and 2) Engine Yard or its equivalent cloud service provider for Rails deployment in the cloud (GitHub, 2013; Engine Yard, 2013), is being evaluated. Second, the development of Rails builds on an Ubuntu Virtual Machine that facilitates students with either PCs or Macs to access the VM. This achieves a degree of independence from a specific hypervisor.

ACKNOWLEDGMENT

The authors are grateful to the National University administration, staff and faculty for providing support for virtual laboratories. The National University faculty among the authors appreciate the vision, collaboration and support provided pro bono by Miles D. Romney, Managing Partner of Spork Labs, Ltd. regarding Rails, virtualization and cloud computing technologies as they apply to teaching. This ongoing work of a decade would not have been possible without his generous collaboration and contribution.

REFERENCES CITED

Agile (2013). Agile Software Development, Retrieved December 28, 2013 from http://en.wikipedia.org/wiki/Agile_software_development

Agile Manifesto (2001). Retrieved August 5, 2013 from http://agilemanifesto.org/February2001

Anderson, R.C. & Romney, G.W. (2013).Comparison of Two Virtual Education Labs –Closing the Gap Between Online and Brick-and-Mortar Schools, *IEEE ITHET2013Conference*, Antalya, Turkey,IEEE Xplore 10.1109/ITHET.2013.6671035

AppFog (2013).PaaS Ruby and Rails Cloud Provider, Retrieved May 2013 from https://www.google.com/#q=appfog

Basecamp (2013). Millions of people in over 180 countries use Basecamp, Retrieved November 13, 2013 from https://basecamp.com/customers

Bittman (n.d.). Gartner Group, Retrieved2012 from http://www.gartner.com/technology/symposium/orlando/hot_topic_bittman.jsp.

CNSS (n.d.).National Centers of Academic Excellence in IA Education(CAE/IAE) Criteria for Measurement. Retrieved August 22, 2013 from http://www.nsa.gov/ia/academic_outreach/nat_cae/cae_iae_program_criteria.shtml

Dey, P., Gatton, T., Amin, M., Wyne, M., Romney, G., Farahani, A., & Cruz, A., (2009).Agileproblem Driven Teaching in Engineering, Science and Technology.*ASEE/PSW-2009 Conference*, San Diego, CA.

Dey, P., Romney, G., Amin, M., Sinha, B.,Gonzales, R., Farahani, A. & Subramanya,S.R. (2012). A Structural Analysis of Agile Problem Driven Teaching, *National University Journal of Research in Innovative Teaching*, Vol. 5, (pages 89–105)

Dreamhost (2012). Web hosting provider PaaS, Retrieved December 13, 2012 from www://dreamhost.com

Engine Yard (2013). PaaS Rails Cloud Provider, Retrieved December 2013 from https://www.engineyard.com/

Fernandez, O. (2008, March 19).Big name companies using Ruby on Rails.Obie Fernandez's blog. Retrieved September 09, 2009 from http://blog.obiefernandez.com/content/2008/03/big-name-compan.html

Fisher, J. (2013).Exploit Code, Metasploit Out for Ruby on Rails Flaws, Retrieved December 9, 2013 from http://threatpost.com/exploit-code-metasploit-module-out-ruby-rails-flaws-011013/773899

Gartner (2007).Findings: The Ruby Language Will Reach 4 Million Programmers by 2013, Retrieved October 2013 from https://www.gartner.com/doc/555609

GitHub (2013). PaaS Code Development Collaboration,Retrieved December 2013 from https://github.com/

Gonzales, R., Romney, G., Bane, C. & Juneau, P. (2012). Cyber Warriors for Cyber Security and Information Assurance, *WASET Conference publications*,Stockholm, Sweden, July 2012

Gonzales, R., Romney, G., Bane, C. & Juneau, P. (2012). Virtual Education Test bedExperimentation, *Journal of Applied Learning Technologies*

Grossman J. (2003) Whitehat Security at Blackhat 2003, "Challenges of Automated Web Application Scanning, Retrieved November 27, 2013 from http://www.blackhat.com/presentations/bh-federal-03/bh-fed-03-grossman-up.pdf

Kali Linux (n.d.). Kali Linux. Retrieved August 2, 2013 from www.kali.org

Kazanjy, P. (2013). How to Source and Recruit Ruby on Rails Developers Using Working With Rails, Retrieved December 20, 2013 from http://www.sourcecon.com/news/2013/08/13/how-to-source-and-recruit-ruby-on-rails-developers-using-working-with-rails/

Lanoy, A. & Romney, G.W. (2006). The Effectiveness of a Virtual Honeynet as aTeaching Resource, *IEEE ITHET 2006 Conference*, Sydney, Australia. http://csrc.nist.gov/publications/nistpubs/800- 125/SP800-125-final.pdf

Matsumoto, Y. (2000, June 12). The Ruby programming language. Pearson Education, InformIT Retrieved June 14, 2009 from www.informit.com/articles/printerfriendly.aspx?p=18225

Metasploit (2013).Penetration Testing Software, Retrieved December 2013, http://www.metasploit.com/

Modis, (2013). Ten Years Old Ruby on Rails: A Look into a Fast-Growing Programming Style, Retrieved October 24, 2013 from http://blog.modis.com/job-seekers/ruby-on-rails-career/, October 1, 2013

Mornini, T. (2011). Here's Why Ruby On Rails Is Hot, Retrieved December 20, 2013 from http://www.businessinsider.com/heres-why-ruby-on-rails-is-hot-2011-5

NSA.gov. (n.d.). National Centers of Academic Excellence in IA Education (CAE/IAE)Criteria for Measurement. Retrieved August 22, 2013 from http://www.nsa.gov/ia/academic_outreach/nat_cae/cae_iae_program_criteria.shtml

NUCSIA.nu.edu. (n.d.). Cyber Security and Information Assurance. Retrieved August 23, 2013 from http://community.nu.edu/csia

Rails Releases(2013).History of Rails Releases, Retrieved December 2013 from http://weblog.rubyonrails.org/releases/, http://en.wikipedia.org/wiki/Ruby_on_Rails

Romney, G.W. & Stevenson, B.R. (2004). An Isolated, Multi-platform Sandbox forTeaching IT Security Engineers, *SIGITE 2004 Conference Proceedings*, ACM Press ISBN: 1-58113-936-5

Romney, G.W., Higby, C., Stevenson, B.R. & Blackham N. (2004), A Teaching Prototype for Educating IT Security Engineers in Emerging Environments, *IEEE ITHET 2004 Conference*, Istanbul, Turkey, IEEE Catalog Number: 04EX898C; ISBN 0-7803-8597-7

Romney, G.W., Gatton, T.M., Cruz, A.P.& Kennedy, P.A. (2008). Integration of ServicesComputing Curricula in Information Technology, *IEEE International Conference on Services Computing*, Honolulu, Hawaii, ISBN: 978-0-7695-3286-8

Romney, G.W.(2009). The Integration of Ruby on Rails as an Agile Teaching Tool in ITCurricula.*ASEE/PSW-2009 Conference*, San Diego, CA

Romney, G.W. & Juneau, P.D. (2009). Implementation of Efficient Two-FactorAuthentication for University Computer Systems, *ASEE-PSW 2009 Conference*, San Diego, CA March 19-20, 2009

Romney, G.W. & Juneau, P.D. (2010). Service-Oriented Architecture and Scalable Two-Factor Authentication for Academic and Medium-sized Enterprises,*IEEE ITHET 2010*, Cappadocia, Turkey

Romney, G.W., Dey, P.P., Amin, M. & Sinha, B.R. (2013). The Flexibility, Agility andEfficiency of Hypervisors in Cyber Security Education, *IEEE ITHET 2013 Conference*, Antalya, Turkey, IEEE Xplore 10.1109/ITHET.2013.6671036

Sahli, M.A. & Romney, G.W. (2010). Agile Teaching: A Case Study of Using Ruby to TeachProgramming Language Concepts, *National University Journal of Research in Innovative Teaching*, La Jolla, CA Volume 3, Issue 1, p. 63

SANS (2013). The Critical Security Controls, Retrieved October 23, 2013 from http://www.sans.org/critical-security-controls/

Scott, M. L. (2006). Programming language pragmatics (2nd ed.). San Francisco: MorganKaufmann Publishers

Security Guide, (2013). Rails Security Guide, Retrieved November 15, 2013 fromrubyonrails.org (2013). Ruby on Rails Security Guide,http://guides.rubyonrails.org/security.html

Sharing (2013). Business, Industry and Academia: Networks and Information Sharing Retrieved December 23, 2013 from http://www.uniheidelberg.de/research/transfer/networks/,

Stevenson, B.R. & Romney, G.W. (2004). Teaching Security Best Practices byArchitecting and Administering a Security Lab, *SIGITE 2004 Conference Proceedings*, ACM Press ISBN: 1-58113-936-5

Thomas, D., Fowler, C. & Hunt, A. (2004). Programming Ruby: The pragmatic programmers'guide . Indianapolis: Addison-Wesley Professional

Thomas, D., Hansson, D., Breedt, L, Davidson, J., Schwarz, A., Gehtland, J. & Clark, M. (2006).*Agile Web Development with Rails, 2nd Edition,* The Pragmatic Bookshelf, ISBN: 978-0-9776-1663-3, Frisco, TX 75033 2006

Veenstra, E. (2007). RubyScript2Exe Project.RubyForge. Retrieved June 17, 2009 from http://rubyforge.org/projects/rubyscript2exe

Weber, H. (2013).Critical Rails vulnerabilities discovered, lets attackers bypass authentication, perform DoS attack, http://thenextweb.com/insider/2013/01/08/critical-rails-vulnerabilities-discovered-lets-attackers-bypass-authentication-systems-perform-ddos-attacks/#!q0WT0.

AUTHORS

Gordon W. Romney (gromney@nu.edu) is a computer science/electrical engineering professor at National University in San Diego. For 20 years he has focused on Cyber Security-IA (CSIA). He created six patents for e-commerce applications of PKI in CSIA, a seminal patent for 3D Computer Graphics (firstrendering.com) and recently the Virtual Instruction Cloud. Of note is his digitally signing all 10K+ fragments of the Dead Sea Scrolls in collaboration with BYU, Claremont University, and the Israel Antiquities Authority. He is a Certified Ethical Hacker and Senior Member of IEEE. He is the architect of an MS CSIA program that is designated a CAE-IAE by NSA/DHS.

Miles D. Romney (miles@sporkapps.com) is managing partner at Spork Labs Ltd., an incubator and tech-services provider. He founded Radiate Media, named Utah's Fastest Growing Company two years running, and Yekra, a next-generation film distributor. He has published with and been invited to present at technology publications and conferences, including IEEE, ICSTC, SIGGRAPH, BIA/Kelsey's ILM, VES, FMX, and Animation Magazine. His work has been recognized by Gartner, Apple, the U.S. Army and others, and featured in The Wall Street Journal, and elsewhere. His collaborators have included Stanford University, National University, the United Nations, Electronic Arts, Adidas, and PAC-12.

Bhaskar Sinha (bsinha@nu.edu) received his PhD in electrical engineering from University of California, Davis, California. He is a professor in the Department of Computer Science, Information and Media Systems, School of Engineering, Technology, and Media, National University, San Diego, California. Sinha has more than 25 years of research and teaching experience in industry and academia. His research interests are in the areas of computer architecture, computer science, digital systems, information technology, and management information systems.

Pradip P. Dey (pdey@nu.edu) received his interdisciplinary PhD and MSE in computer and information sciences from University of Pennsylvania, Philadelphia. He serves as a professor at the Department of Computer Science, Information and Media Systems, School of Engineering, Technology, and Media, National University, San Diego, California. His major research interests are computational models, linguistics, mathematical reasoning, software engineering, user interface, visualization, teaching-learning methods, and communication.

Mohammad N. Amin (mamin@nu.edu) is a professor at National University. He received his PhD and MS in electrical engineering and computer engineering, and MS in solid state physics from Marquette University, Milwaukee, Wisconsin. He also received a M.Sc. and B.Sc. in physics from Dacca University, Bangladesh. He joined National University in 1998 as an assistant professor and developed a new MS program in wireless communications in 2004. He has published and presented more than 90 papers, three U.S. patents, and edited nine books/proceedings. He received an R&D award in 1996 for his outstanding research contributions.

Assessing Security Against a Framework: Wireless Local Area Networks in a Classified Environment

Aftab Ahmad, PhD | Ping Wang, PhD

ABSTRACT

In this paper, we present an analysis of security assessment of Wireless LANs (WLANs) in a classified environment. The analysis is based on a technique derived from ITU Recommendation X.805. We first assess the Standard IEEE 802.11 and find it wanting in many ways to receive as assessment value of about 31%. However, a WLAN in a classified environment in compliance with Department of Defense (DoD) Instruction 8420.01 is robust. The Recommendation X.805 classifies attacks on network in five threat categories, and we assess each category, as well as the overall security. We determine the security measure for the cases of a compromised and uncompromised environment and find that even a compromised environment provides more security by implementing DoD 8420.01 than the Standard IEEE 802.11 (the Standard) itself.

OVERVIEW AND PROBLEM STATEMENT

In this paper, the authors employ the security assessment mechanism proposed by Ahmad (2011) on the United States Department of Defense (DoD) Guidelines 8420.10. The Guideline suggests that IEEE802.11 deployment can be regarded as safe in a classified environment as long it meets certain conditions as set forth in this document and the ones referenced in the Guideline. The proposed assessment mechanism in the study by Ahmad (2011) is primarily derived from the ITU X.805 Recommendation. It places the onus of securing a system on eight security measures, called *dimensions*. A subset of dimensions takes care of one of the five *threat categories* defined by the framework. In order to protect a system completely, a system has to be protected at infrastructure, service and applications *layers*. Data at all these layers from all types of activities, user data exchange, control activities and management activities have to be protected. The DoD 8420.10 WLAN has some add-ons to the Standard security features, as outlines in (Ahmad, 2010). The add-ons are in line with the other security related best practices at DoD. When IEEE 802.15.4 was assessed in Afolabi, Ahmad, and Kim (2010) against X.805, it was found to be way below an acceptable value. It only makes sense that the Standard IEEE 802.11 LANs and the IEEE 802.11 LANs installed with DoD 8420.10 should also be analyzed on the same scale. We do that in this paper.

Security Assessment

Work on network security frameworks has been slow but progressive. Due to the lack of a standard unit for measuring security, comparison to a framework or a guideline is the only choice. Among the frameworks, the following are prominent contributors: the National Institute of Standards and Technology's (NIST) Risk Management Framework (RMF) (Arrington, 2013), Lucent's Security Framework (McGee, A. R. at el, 2004), which resulted in the International Telecommunications Union's (ITU) Recommendation X.805 (McGee, Chandrashekhar, & Richman, 2004), and Cisco's Integrated IT Security Framework (Cisco, 2006), and the Internet Security Protocol (IPsec) (Cebula, 2011). The IPsec provides security architecture for the Internet and has been analyzed as a framework in **(**Arrington, 2013). Of these, IPsec is mostly used to provide options for Internet data in transit only. It has no component for just storing the data, even though the storing can be considered as transmission between a computer and a storage device and same security concept can be applied. However, this will result in unnecessary overhead, something that may make IPsec not the ideal suite for a framework. Cisco considers business aspects as crucial, which is a strength and a weakness; if the goal is to be all-encompassing, it is the former, but if the goal is to have a measure of security then it is the later. The NIST RMF is pretty focused and yet comprehensive. It has the phases to go through the complete cycle of design, deployment, operation and monitoring. The design is based on impact-oriented information categorization, which is akin to risk based design. Even the operation phase employs risk as a central concept, that can justify drawing parallels between Cisco and NIST frameworks. The NIST framework can be used to measure security in a mathematical framework that is derived from risk analysis (NIST, 2010). The threat categorization in these frameworks is from the risk point of view. However, in the Lucent Network Security Framework, hence forward ITU X.805, all threats have to belong to one of the five categories and a set of measures is assumed to take care of all threats in a given category (see later for more details). This is helpful in developing a mathematical framework for assessing security of any system, hardware, software, application or communications. That is why this is a preferred approach and has been taken in (Ahmad, 2011) and expended here further to include WLANs in a classified environment. Following is the layout of the rest of the paper. In the next section, we will have a brief discussion on the security frameworks described above. This discussion is based on Arrington (2013).

AHMAD'S ASSESSMENT TECHNIQUE

In the proposal by Ahmad (2010), an assessment technique based on probabilistic modeling of X.805 has been suggested. We will reproduce this technique from Ahmad (2010) in the following.

Figure 1 shows a map of security dimensions and their relation to threat categories as per the X.805. From this figure, we can represent security against each threat as an eight-element vector showing the need of each dimension or lack of it (a more rigorous discussion is given later). For example, the security vector for Disclosure would be (1,1,1,1,1, 0, 1,0), where the left-most '1' means that access control is required from Figure 1 and the right most '0' means that the privacy dimension is not required.

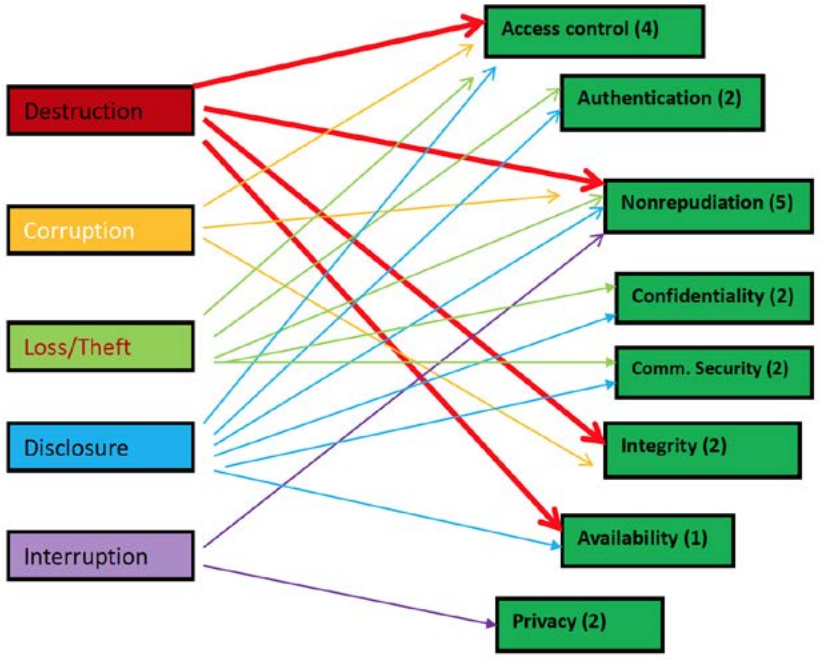

FIGURE 1. DEPENDENCE OF THREATS ON DIMENSIONS

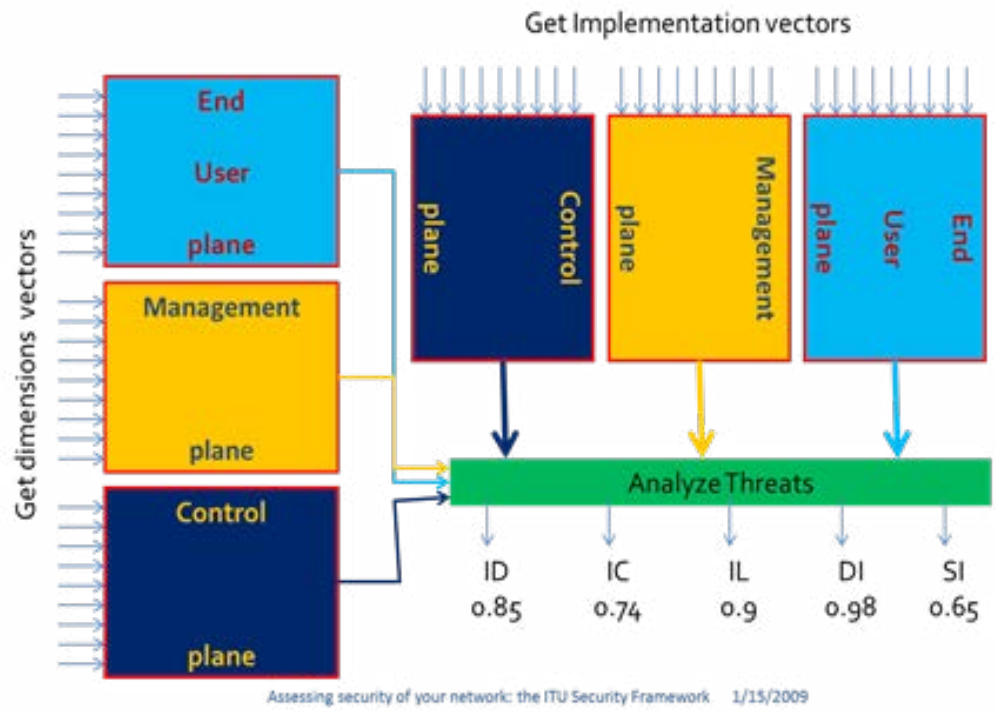

FIGURE 2. CONCEPTUAL SECURITY ASSESSMENT MODEL

These vectors together with the corresponding implementation vectors (see Figure 2 and discussion later) determine the raw security system. In order to determine a single number representing the assessed amount of security, each threat needs to be analyzed in terms of the impact of the implementation on the corresponding threats. Figure 2 shows this concept in which the security assessment system comes up with numbers for each threat type depending on the dimension vectors and the implementation vectors (see below the definitions). In current systems, the dimension vectors can be traced (effectively what Lucent's approach does). There is not a substantial amount of work available in allocating implementation vectors. The implementation vector would actually be a measurement of how secure a dimension is on each of the three security layers.

From Figure 1, we know that each dimension affects security against certain threat types. In the following, we define the terms introduced in the model.

A. Dimension Vector (V_{DV})

The Dimension Vector (V_{DV}) of a security system in general indicates whether a dimension is implemented or not. It consists of eight elements, each having a value of '1' if the corresponding dimension is implemented or '0' if not implemented. The left-most element represents 'access control' and the right most 'privacy'. The order between 'access control' and 'privacy' follows from Fig. 1. At a glance, the V_{DV} of a network, device or a protocol layer provides quick information of the extent of implementation.

B. Weight Vector (V_{WV})

The Weight Vector is an eight-digit (non-binary in general) vector that shows the security impact of each dimension. In this paper, it is assumed for simplicity that all dimensions have the equal impact on a threat for which they are required. We arbitrarily choose a number that shows the number of threats that are affected by the implementation of the corresponding dimension. The leftmost digit is for 'access control' (corresponding to DV). We use the notation V_{WV} to denote the weight vector. As sees in Fig. 1, access control impacts information destruction, information corruption, information loss/theft, and information disclosure. So, it's assumed to have a weight of 4. More research is required in defining and determining the weight vectors for a given implementation of each dimension. With the assumptions of this paper, the V_{WV} should be {4, 2, 5, 2, 2, 2, 2, 1} or a fully secure system, as seen from Fig. 1.

C. Threat Vector (V_{TH})

Threat vectors show the dependence of protection against a threat category considering all eight dimensions. The X.805 recommendation defines the threat vectors for each threat category. We use the notation V_{TH} (.) for threat vector. From Fig. 1, we get the following values for the threat vectors. A '1' implies that a dimension is required to protect against a threat and a '0' implies that the corresponding dimension is not required.

Threat vector for Information destruction:

V_{TH} (ID): (1,0,1,0,0,1,1,0)

Threat vector for Information corruption

V_{TH} (IC): (1,0,1,0,0,1,0,0)

Threat vector for Information removal/loss/theft:

V_{TH} (IR) = (1,1,1,1,1,0,0,0)

Threat vector for Disclosure of information:

V_{TH} (DI) = (1,1,1,1,1,0,0,1)

Threat vector for Service interruption:

V_{TH} (SI) = (0,0,1,0,0,0,0,1)

The leftmost value shows dependence on 'access control' and the right-most on 'privacy', etc.

It should be pointed out here that an alternative framework can be designed by appropriately changing the threat vectors for the same implementation of dimensions.

D. Security Implementation Vector (V_{SIV})

Finally, the security implementation vector ($\boldsymbol{V}_{SIV}$) shows the security provided by actual implementation of dimensions in a system, layer or a device. For example, a value of (1,1,1,1,1,1,1,1) shows that all the eight security dimensions have been implemented to provide an impact of 100%, while a value of (0,0,0,0,0,0,0,0) shows that none of them is implemented. The leftmost value is for 'access control; while the rightmost for 'privacy' according to Fig. 1. For this paper, the security implementation vector is the same as the dimension vector. Once research about the comparative strengths of various implementations (or algorithms) of a dimension is matured, $\boldsymbol{V}_{SIV}$ will represent the strength of implementation of a dimension. For example $\boldsymbol{V}_{SIV}$ = $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8\}$ means that access control implementation provides a security impact equal to α_1 and privacy implementation provides a security equal to α_8 and so on. The values of α_k's are assumed to vary between 0 and 1 inclusive. It may be noted that every plane on every layer will have a different value of $\boldsymbol{V}_{SIV}$ in general. Additionally, each threat category can have its own $\boldsymbol{V}_{SIV}$ value. The difference between the weight vector and implementation vector is that the former relates to the impact of a dimension on the overall system security while the later relates to its implementation strength in comparison with other implementations. For example, the weight vector for data confidentiality tells us how many threats the system will be exposed to in the absence of data confidentiality, while its implementation vector will tell how strong the algorithm is in implementation. This is also an area open for further research.

E. Security Assessment Model

Let S_i be the security against a threat 'i' and ω_i denote the impact of this threat on the overall system security, where i has a value from among (ID, IC, IR, DI, SI) depending on threat category.

Then, following from the above definitions of various vectors, we define the security against threat *'i'* by the following relations:

Defining $\boldsymbol{P}(\boldsymbol{a},\boldsymbol{b}) = \{a_i b_i\}$ as the Hadamard product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. The Hadamard (also, Schur product) is the products of two matrices such that each element of the resulting matrix is the product of corresponding elements of the operand matrices. It can be proven that:

$$\boldsymbol{P}(\boldsymbol{a},\boldsymbol{b}) = [\delta_{ij}\{[\boldsymbol{a}^T\boldsymbol{b}][\mathbf{1}^T]\}]^T \quad \ldots(1)$$

Where,

δ_{ij} is the Kronecker's delta function defined as:

$\delta_{lm} = \{l = m\}$ meaning that $\delta_{lm} = 1$ when $l = m$ and zero otherwise.

[**1**] is a row vector of eight 1's

$\boldsymbol{x}^T$ is the transpose of $\boldsymbol{x}$

Using the definitions of various vectors, we define the security S_i provided against the threat 'i' as follows:

$$S_i = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}} = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}}_i \quad \ldots (2)$$

A dot '.' between two vectors denotes the dot product or scalar product and the absence of a dot indicates a matrix multiplication

F. Interpretation of Equation (2)

Equation (2) is the ratio of the total weights implemented in all dimensions relating to thwarting threat i, to the total weights necessary for threat i in order to conform to ITU X.805. As a check, we see that for a full implementation of dimensions against a threat, the numerator is equal to the denominator providing 100% protection in accordance with the X.805 standard.

If we define the vector $\boldsymbol{S} = \{S_i\}$ with elements that denote the security against each of the five threats, and the vector $\omega = \{\omega_i\}$ the impact vector whose elements are the impact of each of the threat category on the overall system security, then the overall system security S can be defined as

$S = \omega.\boldsymbol{S}$ …(3)

$\omega = (\omega_{ID}, \omega_{IC}, \omega_{IR}, \omega_{DI}, \omega_{SI})$

$\boldsymbol{S} = (S_{ID}, S_{IC}, S_{IR}, S_{DI}, S_{SI})$

The dot product of Equation (3) can be expanded to the following:

$S = \omega_{ID} S_{ID} + \omega_{IC} S_{IC} + \omega_{IR} S_{IR} + \omega_{DI} S_{DI} + \omega_{SI} S_{SI}$

G. Ideal Case Scenario

Equation (2) defines the security measure against a threat category. For an ideal case, we will have the following values of various vectors for ID.

$\boldsymbol{V}_{TH}$ (ID): {1,0,1,0,0,1,1,0}

$\boldsymbol{V}_{WV}$ (ID): {4,2,5,2,2,2,1,2}

$\boldsymbol{V}_{SIV}$:{1,1,1,1,1,1,1,1}

$\boldsymbol{V}_{WV}(ID \,.\boldsymbol{V}_{TH}(ID) = 4 + 5 + 2 + 1 = 12$

$P(\boldsymbol{V}_{SIV}. \boldsymbol{V}_{TH}) = \{1,0,1,0,0,1,1,0\}$

$P(\boldsymbol{V}_{SIV}. \boldsymbol{V}_{TH}).\boldsymbol{V}_{WV}(ID) = 4+5+2+1 = 12$

From Equation (2) for this case:

$$S_{ID} = S_i = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}} = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}}_{ID}$$

$= 1.0 = 100\%$

Similarly, it is easily shown that for an ideal case, the overall security is 100% from Equation (3).

Equations (1)-(3) provide a model for labeling a system in terms of security with X.805 as a measuring unit.

Values of ω depend on the impact of compromise against each threat category. One way to assign this number is to use risk analysis with a maximum damage Δ done to the system if all categories are vulnerable and $\delta_k$the damage due to category k being exposed, then $\omega_k = \delta_k/\Delta$. It must be noted that the weight vector requires analysis of each dimension and the assumption that its components are equal to the number of threats it thwarts is rather simplistic.

SECURITY IN IEEE 802.11

In this section, we will apply the analytical model of the previous section on IEEE 802.11 Standard (the Standard). Section 8 of the Standard provides various options for access control, key management and distribution, and data confidentiality. We assume that the strongest options, including encryption, are employed for the network under consideration. This assumption can be used to argue that the implementation weights (the α_k's) have a value of unity for all dimensions in the V_{SIV}. We focus only on the user data plane, assuming that the management and control activities are fully secure. We consider robust secure network (RSN) class of security algorithms, CCMP encryption protocol, a server-based trusted and secure access control, key generation, and distribution and regeneration system. We will look at each dimension before considering individual threat categories and the overall security provided in the Standard.

In the following, we denote preceding vectors by a superscript to denote the dimension. Accordingly, ${}^1V_{SIV}$ is the security implementation vector component for access control and ${}^8V_{SIV}$ is the same for privacy.

Analysis of Security Dimensions

Access Control

The IEEE 802.11 2012 edition provides a port-based access control between the access point (AP) and a wireless station (STA). For speedy roaming, there is also pre-authentication provided in the Standard so that a STA could bypass access control and authentication, and be authenticated directly to the AP of an impending WLAN. The access control is provided only for the medium access control (MAC) layer of the protocol stack. Layer 1 (PHY) is unprotected and signals at this layer can be compromised in both directions, from the legitimate STA to an attacking STA and vice versa. That is why the 802.11 WLAN can be jammed easily. Therefore, we assign a value of 1 to dimension vector component for access control (${}^1V_{DV}$). However, the security implementation vector component (${}^1V_{SIV}$) value is less than 1. Let it be ½(1+ α_1), where α_1 is the value

of ${}^1V_{SIV}$ corresponding to the PHY. The multiplying factor ½ implies that there will be equal number of attacks on MAC and PHY layers, as assumption we follow throughout this paper.

Authentication

Authentication is closely tied to access control. In fact, 802.11 or any network must provide a way of authenticating credentials even before access can be provided. However, in this discussion, by authentication we imply the infrastructure that stores the credentials and the protocols used to protect and make use of these credentials. In a centralized server based system, authentication can be made pretty strong and fool-proof. That is, on the MAC layer. Whether a physical signal is an authenticated one or not depends on whether there is some administrative control available to counter an illegitimate transmission or at least identify its location. While similar systems are employed in practice, the Standard does not provide for this. Therefore, we will assume an imperfect implementation while considering the MAC layer to be 100% strong in authentication. Let ½(1+ α_2) be the value of ${}^2V_{SIV}$.

Non-repudiation

Non-repudiation relates to the ability to place a specific person behind an act. It may seem like strong authentication implements non-repudiation as well, but such is not the case in reality. The purpose of authentication is related to allowing connection to the network by only the legitimate users while the non-repudiation is the ability that is not needed every time a user connects to the network. It will be needed when there is a need to confirm a specific act for which the responsible user can't say that s/he couldn't have done it. An example of a legitimate authentication but no non-repudiation is when user accounts are accessible to a trusted party, such as the company president. Such a system does not have non-repudiation against the original account holder because more than one party has access to the same authentication credentials. It does have the non-repudiation against the company because it can be proven that no one outside the company could use those credentials. In other words, non-repudiation is a function of implementation and usually a third party is hired for this. We conclude that the Standard does not have non-repudiation, even though it does provide the means for implementing one in the form of strong authentication and data confidentiality. In the end, non-repudiation depends on how credentials are managed and what kind of proof is there about their management. Let α_3 be the value of ${}^3V_{SIV}$. Its value in the Standard is zero in light of the above discussion.

Confidentiality

In the pre-RSN IEEE 802.11 Standard, weaknesses in initialization vector and random number generator were the primary concerns. In the IEEE 802.11*i* (now part of the standard), they were both rectified. However, encryption is done only at the MAC layer. There is signal scrambling done at the PHY (also follows from MAC), but since the scrambling code is published, it does not cover for encryption, which, for example, could be done through the use of spread spectrum modulation with a secret code. Let ½(1+α_4) be the value of ${}^4V_{SIV}$ in light of the above discussion.

Communications Security

By its very nature, a wireless network does not have communications security unless it is implemented in a way that an attacker can be instantly stopped, be it active or passive. The Standard does not make provisions to protect communications security from being compromised; therefore, we consider, α_5, the value of ${}^5V_{SIV}$ to be zero for the Standard. Since communications security deals with the security of the path of communications, we are justified in this assumption as long as the WLAN RF coverage area is not physically secured.

Integrity

As mentioned above, the current IEEE 802.11 Standard has strong data integrity at the MAC layer. We assign it a value of 1. At the PHY layer, data integrity would require the inability for an intruder

to modify signal in any way. This could be done, for example, by the same system that provides communications security, as explained in §4.5. We consider $\frac{1}{2}(1+\alpha_6)$ be the value of ${}^6V_{SIV}$, whereas α_6 is zero for the Standard.

Availability

Attacks on availability of a WLAN could be thwarted by measures against jamming, against inadvertent radiation in the same band, and by having the capability of location identification and restraining the activities of an attacker. None of these measures is provided in the Standard. One may argue that access control provides a measure against attacks on availability by restricting the use of network to only the legitimate user. However, access control cannot stop a WLAN from being jammed. Even at the MAC layer, an access point buffers for association requests can easily be filled by making simple changes to the request packets. Therefore, we consider α_7, the value of ${}^7V_{SIV}$, to be zero for the Standard.

Privacy

Privacy of signal and data in case of IEEE 802.11 will go hand in hand with confidentiality because there is no routing involved. It has a strong component at the MAC layer and nothing at the PHY. So, we set consider $\frac{1}{2}(1+\alpha_4)$ as the value of ${}^8V_{SIV}$ to be zero in light of this discussion.

Security against Threat Categories

The table below is the summary of security analysis of individual dimensions.

Dimension	Value in IEEE 802.11
Access control	${}^1V_{SIV} = \frac{1}{2}(1+\alpha_1)$, $\alpha_1 = 0$ in the Standard
Authentication	${}^2V_{SIV} = \frac{1}{2}(1+\alpha_2)$, $\alpha_2 = 0$ in the Standard
Non-repudiation	${}^3V_{SIV} = \alpha_3 = 0$ in the Standard
Confidentiality	${}^4V_{SIV} = \frac{1}{2}(1+\alpha_4)$, $\alpha_4 = 0$ in the Standard
Communication Security	${}^5V_{SIV} = \alpha_5 = 0$ in the Standard
Integrity	${}^6V_{SIV} = \frac{1}{2}(1+\alpha_6)$, $\alpha_6 = 0$ in the Standard
Availability	${}^7V_{SIV} = \alpha_7 = 0$ in the Standard
Privacy	${}^8V_{SIV} = \frac{1}{2}(1+\alpha_4)$, $\alpha_4 = 0$ in the Standard

To remain consistent with the assumptions of this paper, the weight vector remains unchanged to:

$$\boldsymbol{V}_{WV} = \{4, 2, 5, 2, 2, 2, 2, 1\} \qquad \ldots 4$$

From Table above, the $\boldsymbol{V}_{SIV} = \{{}^1V_{SIV}, {}^2V_{SIV}, {}^3V_{SIV}, {}^4V_{SIV}, {}^5V_{SIV}, {}^6V_{SIV}, {}^7V_{SIV}, {}^8V_{SIV}\}$

$$\boldsymbol{V}_{SIV} = \{ \tfrac{1}{2}(1+\alpha_1), \tfrac{1}{2}(1+\alpha_2), \alpha_3, \tfrac{1}{2}(1+\alpha_4), \alpha_5, \tfrac{1}{2}(1+\alpha_6), \alpha_7, \tfrac{1}{2}(1+\alpha_4)\} \qquad \ldots 5$$

For the Standard specification, we substitute α_7 with zeros and get:

$$\boldsymbol{V}_{SIV} = \{ \tfrac{1}{2}, \tfrac{1}{2}, 0, \tfrac{1}{2}, 0, \tfrac{1}{2}, 0, \tfrac{1}{2}\} \qquad \ldots 6$$

The threat vectors remain unchanged:

V_{TH} (ID): (1,0,1,0,0,1,1,0) …7

Threat vector for Information corruption

V_{TH} (IC): (1,0,1,0,0,1,0,0) …8

Threat vector for Information removal/loss/theft:

V_{TH} (IR) = (1,1,1,1,1,0,0,0) …9

Threat vector for Disclosure of information:

V_{TH} (DI) = (1,1,1,1,1,0,0,1) …10

Threat vector for Service interruption:

V_{TH} (SI) = (0,0,1,0,0,0,0,1) …11

For Information Destruction (ID),

$P(V_{SIV}, V_{TH}).V_{WV}$ = P({ ½, ½, *0*, ½, *0*, ½, *0*, ½}, {1,0,1,0,0,1,1,0}) = {*½, 0, 0, 0,0, ½, 0, 0*}.{*4, 2, 5, 2, 2, 2, 2, 1*}

= *3* …12(a)

For Information Corruption (IC),

$P(V_{SIV}, V_{TH}).V_{WV}$ = P({ ½, ½, *0*, ½, *0*, ½, *0*, ½}, {1,0,1,0,0,1,0,0}) = {*½, 0, 0, 0,0, ½, 0, 0*}.{*4, 2, 5, 2, 2, 2, 2, 1*}

= *3* …12(b)

For Information Removal (IR),

$P(V_{SIV}, V_{TH}).V_{WV}$ = P({ ½, ½, *0*, ½, *0*, ½, *0*, ½}, {1,1,1,1,1,0,0,0}) = {*½, ½, 0, ½,0, 0, 0, 0*}.{*4, 2, 5, 2, 2, 2, 2, 1*}

= *4* …12(c)

For Disclosure of Information (DI),

$P(V_{SIV}, V_{TH}).V_{WV}$ = P({ ½, ½, *0*, ½, *0*, ½, *0*, ½}, {1,1,1,1,1,0,0,1}) = {*½, ½, 0, ½,0, 0, 0, ½* }.{*4, 2, 5, 2, 2, 2, 2, 1*}

= *4.5* …12(d)

For Service Interruption (SI),

$P(V_{SIV}, V_{TH}).V_{WV}$ = P({ ½, ½, *0*, ½, *0*, ½, *0*, ½}, {0,0,1,0,0,0,0,1}) = {*0, 0, ½, 0,0, 0, 0, ½*}.{*4, 2, 5, 2, 2, 2, 2, 1*}

= *3* …12(e)

$V_{TH}(ID).V_{WV}$ = {*1,0,1,0,0,1,1,0*} . {*4, 2, 5, 2, 2, 2, 2, 1*} = *13*

$V_{TH}(IC).V_{WV} = \{1,0,1,0,0,1,0,0\} . \{4, 2, 5, 2, 2, 2, 2, 1\} = 11$

$V_{TH}(IR).V_{WV} = \{1,1,1,1,1,0,0,0\} . \{4, 2, 5, 2, 2, 2, 2, 1\} = 15$

$V_{TH}(DI).V_{WV} = \{1,1,1,1,1,0,0,1\} . \{4, 2, 5, 2, 2, 2, 2, 1\} = 16$

$V_{TH}(SI).V_{WV} = \{0,0,1,0,0,0,0,1\} . \{4, 2, 5, 2, 2, 2, 2, 1\} = 6$

$$S_{ID} = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}} = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}}_{ID} = 3/13 \quad \ldots 13(a)$$

$$S_{IC} = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}} = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}}_{ID} = 3/11 \quad \ldots 13(b)$$

$$S_{IR} = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}} = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}}_{ID} = 4/15 \quad \ldots 13(c)$$

$$S_{DI} = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}} = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}}_{ID} = 9/32 \quad \ldots 13(d)$$

$$S_{SI} = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}} = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}}_{ID} = 1/2 \quad \ldots 13(e)$$

Overall Security of the IEEE 802.11 Standard

From Equation (3), we can write:

$\omega = (\omega_{ID}, \omega_{IC}, \omega_{IR}, \omega_{DI}, \omega_{SI}) = \{1/5, 1/5, 1/5, 1/5, 1/5\}$

$\boldsymbol{S} = (S_{ID}, S_{IC}, S_{IR}, S_{DI}, S_{SI})$

The dot product of Equation (3) can be expanded to the following:

$S = \omega_{ID} S_{ID} + \omega_{IC} S_{IC} + \omega_{IR} S_{IR} + \omega_{DI} S_{DI} + \omega_{SI} S_{SI}$

$= 31\%$ …13(f)

We conclude from Equation (19) that the IEEE 802.11-2012 is secure about 31% with respect to the proposed assessment method based on ITU-X.805.

WLAN IN A CLASSIFIED ENVIRONMENT

Per the DoD 4820.01 Guideline, IEEE 802.11 based WLANs can operate in a classified environment. The guideline sets aside additional requirements as outlined in (Ahmad, 2010). The additional requirements for the WLANs and information assurance measures are reproduced from Table 1 and Table 2 of (Ahmad, 2010).

TABLE 1. WIRELESS LAN REQUIREMENTS FOR CLASSIFIED ENVIRONMENT (DOD2009)

Requirement/Entity	Implementation	Detail
Product Certification	NSA type-1	Key, key management, concepts of operations, interoperability requirements certified separately. Type-1 products approved by NSA Commercial Communication Security (COMSEC) Evaluation Program (CCEP)
Physical Security	Access Points (AP) be tampering detectable	If not secured in COMSTEC-approved security containers, APs be poll-able by serial numbers or MAC addresses, and transmit at the lowest power.
Information Assurance	10 measures	See Table 2
WLAN-enabled Personal Electronic Devices (PEDs)	Certified encryption and physical security of data and PED	NSA-type 1 compliant encryption, Storage media and PED is GSA security container according to (DOD1997)
Wireless intrusion detection system (WIDS)	WIDS on all wired and wireless LANS for monitoring and detection of WLAN policy violations.	For IEEE 802.11 devices only, detect unauthorized devices, locate them and take appropriate action.
Security Technical Implementation Guide (STIG) compliance	Must be compliant to Wireless STIG	(DOD20072)

TABLE 2. MEASURES FOR INFORMATION ASSURANCE OF CLASSIFIED WLANS

Measure	Requirement
Maximum key life	90 days
Maximum session timeout	30 minutes
Identification and authentication (I&A)	National Security Telecommunications and Information Systems Security Instruction No. 1000
Integrity and non-repudiation	
Operations/configurations adjustments	According to the guidance issued by Secure Internet Protocol Routing Network (SIPRNet) Connection Approval Office (CAO). Also written procedures for NSA-type-1 devices and key material.
SIPRNET connection approval package	Must be on file and be updated with CAO to include WLAN
WLAN operation in any sensitive classified information facility (SCIF)	Must be approved by Director Central Intelligence Directive 6/9 or Intelligence Community Directive Number 503
Certified TEMPEST Technical Authority (CTTA) notification	Completed before any installation and operation of WLANs
Certification/accreditation	(DOD2007) and (FIPS20011402)
Client side access control	MAC filtering at the APs

We will discuss in the next section how the above tables impact the security measures of ITU X.805.

Impact of DoD 8420.01 on Security Dimensions

Access Control

The fact that wireless intrusion detection systems (WIDS) are required in order to detect and stop violators renders PHY access control on sound ground. This is in addition to the physical security requirement of having all access points transmit at the lowest power levels. We give access control 1.

Authentication

Several requirements emphasize on authentication and the one for SCIF and Certified TEMPEST Technical Authority (CTTA) notification covers PHY authentication in some way. There is now a STIG requirement specifically for mobile devices (Crowe, 2013). Because of this and the IDS requirement, we give authentication a value of 1.

Non-repudiation

Non-repudiation requires evidence collection and preservation systems. A third party, such as VeriSign, using digital certification makes a good example of such a system. From the information assurance measures, we can see that there is sufficient amount of conditioning to warrant the presence of a non-repudiation system, together made by the WIDS, with these measures. So, we will give it a 100%.

Data Confidentiality

We give 1 out of 1 to confidentiality due to TEMPEST (Telecommunications Electronics Material Protected from Emanating Spurious

Transmissions), which was designed to stop radiation from leaving the monitor. Its application to wireless signal is expected in measures to force signal containment within the area of application.

Communications Security

The TEMPEST also takes care of communications security, as there is no chance for the signal or packet data to take an unknown route. The only vulnerability is due to the very nature of a wireless network that is, being of broadcast type. The way out of this can be by using a secret code for each communications instance. This is a provision not present in the DoD Guideline. Therefore, we give it a value of $\alpha_5 = 1-\varepsilon$. Here, ε is the probability that the environment is compromised and someone has the ability to steal the signal.

Data Integrity

Since the MAC layer of the Standard is fool-proof against attacks on data integrity, it is the PHY layer that requires reinforcements if any. The interfering signal and jamming are ways to tamper with the PHY layer signal and that can't happen on a WLAN set up under the Guideline due to requirement of a WIDS except from within. However, even in a compromised classified environment, tampering of data can only unravel the compromise, so we don't expect this to be the case. Consequently, the data integrity can be described as having a value of unity.

Availability

In an ideal network, availability will have two sides, the network and its resources being available for service and not being available for attacks. However, it is the former that is counted as availability. Based on the fact that there is WIDS operating and the environment is classified, we give it a full 1.

Privacy

Attacks on data integrity, data confidentiality and communications security are facilitated by a lack of privacy. In reality, law and its enforcement is the only true defense against offences on privacy. On a technical level, ensuring that data and signal can reach only those storage and processing devices that are the intended recipients is how a breach of privacy requirements can be thwarted. In a wireless LAN environment under DoD 8420.01, it is possible to ascertain the sanctity of the equipment, system configuration to avoid human errors and fallibility. Thus, keeping in view that a compromised environment can play its role for this dimension too, this dimension is given $\alpha_8 = 1-\varepsilon$. One may notice a difference in the way confidentiality and privacy are related to each other for the actual standard versus the DoD Guideline. This is because a lack of confidentiality at the PHY layer results in a lack of privacy as well, but the presence of a confidentiality measure does not necessarily remove the lack of privacy. In the case of the IEEE 802.11 Standard, there is no provision at the PHY layer against attacks on confidentiality. However, in the case of the DoD Guideline, the TEMPEST keeps the signal to the restricted parties but that does not count as privacy, as the signal can be peeped into by wireless stations that are not its intended recipients.

ASSESSMENT OF DOD 8420.01 WLAN

In this section, we repeat the analysis for the case of DoD Guideline 8420.01 as discussed in §5.

To remain consistent with the assumptions of this paper, the weight vector remains unchanged to:

$\boldsymbol{V}_{WV} = \{4, 2, 5, 2, 2, 2, 2, 1\}$

From Table above, the $\boldsymbol{V}_{SIV} = \{{}^1V_{SIV}, {}^2V_{SIV}, {}^3V_{SIV}, {}^4V_{SIV}, {}^5V_{SIV}, {}^6V_{SIV}, {}^7V_{SIV}, {}^8V_{SIV}\}$

$\boldsymbol{V}_{SIV} = \{ 1, 1, 1, 1, 1-\varepsilon, 1, 1, 1-\varepsilon\}$

$\boldsymbol{V}_{TH}$ (ID): (1,0,1,0,0,1,1,0) …7

Threat vector for Information corruption

$\boldsymbol{V}_{TH}$ (IC): (1,0,1,0,0,1,0,0) …8

Threat vector for Information removal/loss/theft:

$\boldsymbol{V}_{TH}$ (IR) = (1,1,1,1,1,0,0,0) …9

Threat vector for Disclosure of information:

V_{TH} (DI) = (1,1,1,1,1,0,0,1) …10

Threat vector for Service interruption:

V_{TH} (SI) = (0,0,1,0,0,0,0,1) …11

For Information Destruction (ID),

$P(V_{SIV}, V_{TH}).V_{WV}$ = P({ *1, 1, 1, 1, 1*-ε, *1, 1, 1*-ε}, {1,0,1,0,0,1,1,0}).V_{WV}

= {*1, 0, 1, 0,0, 1, 1, 0*}.{*4, 2, 5, 2, 2, 2, 2, 1*}

= 13 …14(a)

For Information Corruption (IC),

$P(V_{SIV}, V_{TH}).V_{WV}$ = P({ 1, 1, *1*, 1, *1*-ε, 1, *1*, 1-ε}, {1,0,1,0,0,1,0,0}).V_{WV}

= {*1, 0, 1, 0, 0, 1, 0, 0*}.{*4, 2, 5, 2, 2, 2, 2, 1*}

= *11* …14(b)

For Information Removal (IR),

$P(V_{SIV}, V_{TH}).V_{WV}$ = P({ 1, 1, *1*, 1, *1*-ε, 1, *1*, 1-ε}, {1,1,1,1,1,0,0,0}).V_{WV}

= {*1, 1, 1, 1, 1*-ε, *0, 0, 0*}.{*4, 2, 5, 2, 2, 2, 2, 1*}

= *15 – 2*ε …14(c)

For Disclosure of Information (DI),

$P(V_{SIV}, V_{TH}).V_{WV}$ = P({ 1, 1, *1*, 1, *1*-ε, 1, *1*, 1-ε}, {1,1,1,1,1,0,0,1}).V_{WV}

= {*1, 1, 1, 1,1*-ε, *0, 0, 1*-ε }.{*4, 2, 5, 2, 2, 2, 2, 1*}

= *16 -3* ε …14(d)

For Service Interruption (SI),

$P(V_{SIV}, V_{TH}).V_{WV}$ = P({ 1, 1, *1*, 1, *1*-ε, 1, *1*, 1-ε}, {0,0,1,0,0,0,0,1}).V_{WV}

= *{0, 0, 1, 0,0, 0, 0, 1-ε}.{4, 2, 5, 2, 2, 2, 2, 1}*

= *6-ε* …14(e)

V_{TH} ***(ID)***.V_{WV} = *{1,0,1,0,0,1,1,0} . {4, 2, 5, 2, 2, 2, 2, 1} = 13*

$$S_{ID} = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}} = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}}_{ID} = 13/13 = 1 \quad \ldots 15(a)$$

$$\boldsymbol{V_{TH}(IC).V_{WV}} = \{1,0,1,0,0,1,0,0\} \,.\, \{4,\ 2,\ 5,\ 2,\ 2,\ 2,\ 2,\ 1\} = 11$$

$$S_{IC} = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}} = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}}_{ID} = 11/11 = 1 \quad \ldots 15(b)$$

$$\boldsymbol{V_{TH}(IR).V_{WV}} = \{1,1,1,1,1,0,0,0\} \,.\, \{4,\ 2,\ 5,\ 2,\ 2,\ 2,\ 2,\ 1\} = 15$$

$$S_{IR} = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}} = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}}_{ID} = 1 - (2/15)\varepsilon \quad \ldots 15(c)$$

$$\boldsymbol{V_{TH}(DI).V_{WV}} = \{1,1,1,1,1,0,0,1\} \,.\, \{4,\ 2,\ 5,\ 2,\ 2,\ 2,\ 2,\ 1\} = 16$$

$$S_{DI} = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}} = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}}_{ID} = 1 - (3/16)\varepsilon \quad \ldots 15(d)$$

$$\boldsymbol{V_{TH}(SI).V_{WV}} = \{0,0,1,0,0,0,0,1\} \,.\, \{4,\ 2,\ 5,\ 2,\ 2,\ 2,\ 2,\ 1\} = 6$$

$$S_{SI} = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}} = \frac{P(V_{SIV}, V_{TH}).V_{WV}}{V_{TH}.V_{WV}}_{ID} = 1 - \varepsilon/6 \quad \ldots 15(e)$$

Assuming equal impact of threat categories:

From Equation (3), we can write:

$\omega = (\omega_{ID}, \omega_{IC}, \omega_{IR}, \omega_{DI}, \omega_{SI}) = \{1/5,\ 1/5,\ 1/5,\ 1/5,\ 1/5\}$

$\boldsymbol{S} = (S_{ID}, S_{IC}, S_{IR}, S_{DI}, S_{SI})$

The dot product of Equation (3) can be expanded to the following:

$$S = \omega_{ID} S_{ID} + \omega_{IC} S_{IC} + \omega_{IR} S_{IR} + \omega_{DI} S_{DI} + \omega_{SI} S_{SI}$$

$$= 1 - (2/15)\varepsilon - (3/16)\varepsilon - \varepsilon/6 = 0.1384 + 0.0615 + 0.0692 = 0.2691 = \quad \ldots 16$$

$$= 1 - 0.4875\ \varepsilon$$

Results

Figure 3 and Figure 4 show results of the analyses in §4 and §6 for the two deployment scenarios discussed in this paper.

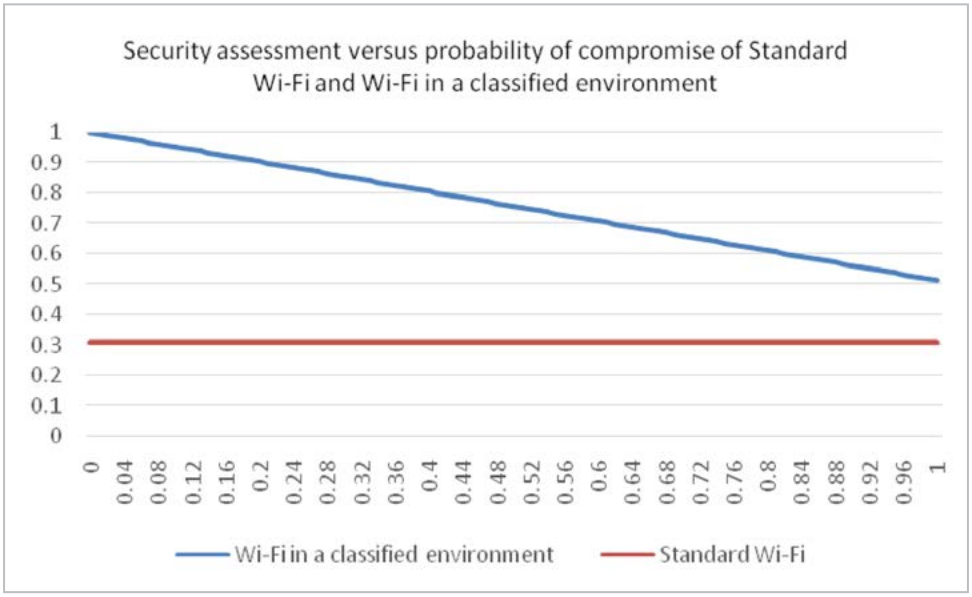

FIGURE 3.

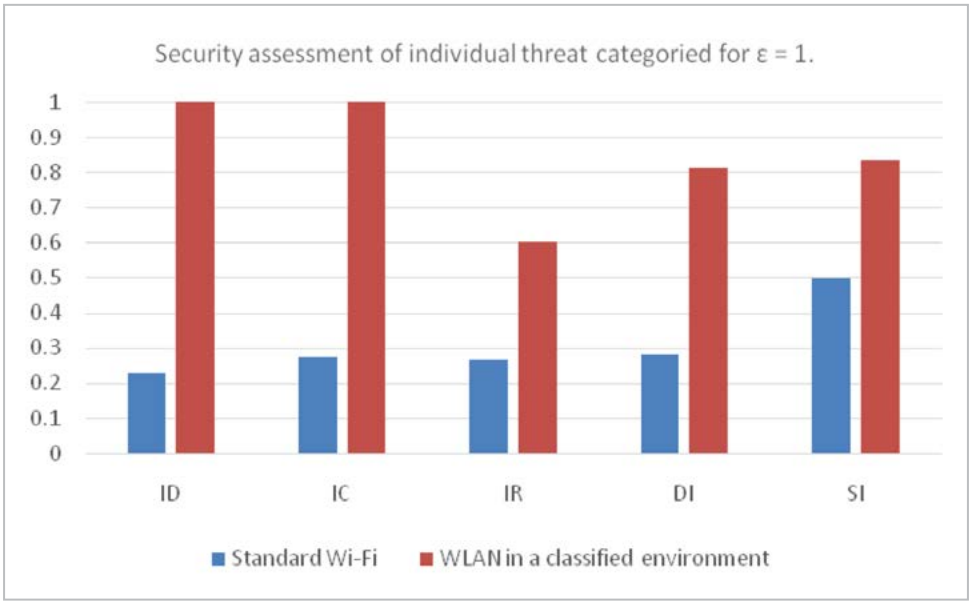

FIGURE 4.

OTHER APPLICATIONS OF WORK – SMART METER

Besides the networking in sensitive areas and financial institutions, the advanced metering infrastructure (AMI) is another area that requires a protection against breaches and attacks that can be assessed properly. In an AMI in which smart meters are connected via a WLAN, an attack within the WLAN can propagate to the smart grid and render the whole energy grid at the mercy of the attacker. Therefore, an analysis that can be used as an indicator of how secure the smart meters are and what are the chances of the WLAN attacks to propagate to the main grid is essential. This work can be applied to such an analysis and is the subject of ongoing research.

CONCLUSIONS AND FUTURE WORK

In this paper, we have applied a security assessment framework to measure security offered by the Standard IEEE 802.11 and the IEEE 802.11-based WLANs in a classified environment. The framework is itself based on ITU X.805 and explained in (Ahmad, 2011). For the WLAN in a classified environment, we employed the additional requirements as suggested by the DoD Guideline 8420.01. The analysis shows that the Standard Wi-Fi provides a level of protection of approximately 31% out of the framework maximum of 100%. However, the WLAN in a classified environment provides very good protection, which would be 100% for an uncompromised environment but still better than the IEEE 802.11 standard even if the environment is compromised. We compare the security against each of the five threat categories as defined by X.805 and find the protection to be 100% for two categories even if the environment is compromised and above 50% in each remaining category even if the environment is 100% compromised. For future follow-up studies, we plan to continue the application work in more situations, such as AMI with smart meters connected via the WLAN. Our future work will also incorporate more variables, such as varied effects of different encryption algorithms, as well as tests to measure the protection effectiveness.

REFERENCES CITED

Afolabi R., Ahmad, A., & Kim, K. (2010). Security assessments of IEEE 802.15.4 standard based on X.805 framework. *International Journal of Security and Networks, 5*(2/3), 188–197.

Ahmad, A. (2010). A note on modeling, simulation and analysis of WLANs in a classified environment. *Proceedings on the Spring Simulation Multi-conference 2010 in the Symposium on Communications Networks*, April 2010, Orlando, FL.

Ahmad, A. (2011). Security assessment of networked system. In D. C. Kar & M. R. Syed (Eds.), *Network security administration and management: Advancing technologies and practices*(pp.115–130), Hershey, PA: IGI Global.

Arrington, M. (2013). On viability of ip security protocol as network security assessment framework. Master's Project Report, Department of Computer Science, Norfolk State University. Spring 2013.

Cebula III, S. L., Ahmad, A., Graham, J. M., Wahsheh, L. A., Williams, A. T., & DeLoatch, S. L. (2011). How secure is WiFi as compared with IPsec? Spring Simulation Multi-conference 2011 (SpringSim2011), April 2011, Boston MA.

Cisco. (2006). Integrated Security Architectural Framework. Retrieved from http://www.ijis.org/docs/CISCO%20integrated%20security%20architecura%20frameworkwhitepaper.pdf

Crowe, G. (2013). DoD okays Blackberries, Android devices, expanding its mobile options. Retrieved from http://gcn.com/articles/2013/05/03/dod-oks-blackberry-android-devices-expanding-mobile-options.aspx

DoD (Department of Defense). (1997). DoD regulation 5200.1-R: Information security program.

DoD. (2007a). DoD instruction 8510.01: DoD information assurance certification and accreditation process (DIACAP).

DoD. (2007b).DoD wireless security technical implementation guide V5R2: Wireless security technical implementation guide.

DoD. (2009). Commercial wireless local-area network (WLAN) devices, systems, and technologies. *Department of Defense Instruction,* Number 8420.01, ASD(NII)/DoD CIO.

McGee, A. R., Chandrashekhar, U., & Richman, S. H. (2004, June). Using ITU-T X. 805 for comprehensive network security assessment and planning. In *Telecommunications Network Strategy and Planning Symposium. NETWORKS 2004, 11th International* (pp. 273–278). IEEE.

McGee, A. R., Vasireddy, S. R., Xie, C., Picklesimer, D. D., Chandrashekhar, U., & Richman, S. H. (2004). A framework for ensuring network security. *Bell Labs Technical Journal, 8*(4), 7–27.

NIST (National Institute of Science and Technology). (2010). Guide for applying the risk management framework to federal information systems (NIST Special Publication 800–837, Revision 1).

AUTHORS

Aftab Ahmad, PhD (aahmad@nsu.edu) (Member ACM, Member NYAS, past Senior Member IEEE), is an associate professor in the Computer Science Department at Norfolk State University. At NSU, he teaches courses on computer architecture, wireless networking, and computer graphics. Ahmad's research interests include wireless networking, subneural brain signaling, and networks of implantable sensors. He has authored two books, a book chapter, and scores of peer-reviewed articles published in international research journals and conference proceedings. Ahmad is a Certified Ethical Hacker (CEHv6). He also teaches a graduate course on cybersecurity at University of Maryland University College as an adjunct. He is an independent consultant for Acacia Research.

Ping Wang, PhD (ping.wang@faculty.umuc.edu), is a professor of Cybersecurity Technology at the Graduate School of University of Maryland University College, where he has been teaching courses in network security and digital forensics and serves as the director of the master's program in cybersecurity. Wang holds a master's degree in computer information science and a PhD in information systems with specialization in e-commerce security risks and decisions. He is a Certified Information Systems Security Professional (CISSP) with consulting and development experience in cybersecurity and information assurance.


EXCELSIOR COLLEGE
FOUNDED IN 1971