8th Annual Symposium on Information Assurance (ASIA '13)

General ASIA Chair:

Sanjay Goel

Information Technology Management, School of Business

University at Albany, State University of New York

Academic Track of 16th Annual NYS Cyber Security Conference

Empire State Plaza Albany, NY, USA

June 4-5, 2013

Proceedings of the 8th Annual Symposium on Information Assurance / Secure Knowledge Management – A Workshop
Academic track of the 16th Annual 2013 NYS Cyber Security Conference
June 4-5, 2013, Albany, New York, USA.

Organizing Committee

Sanjay Goel, General Chair
Director of Research, NYS Center for Information Forensics and Assurance
Associate Professor, Information Technology Management, School of Business, University at Albany, SUNY

H. Raghav Rao, ASIA Chair
SUNY Distinguished Service Professor
Department of Management Science and Systems, University at Buffalo, SUNY

Tae (Tom) Oh, Program Chair
Associate Professor, Dept. of Information Sciences and Technologies, Dept. of Computing Security, Rochester Institute of Technology

Bill Stackpole, Review Co-Chair
Associate Professor, Dept. of Computing Security
Rochester Institute of Technology

Sylvia Perez-Hardy, Review Co-Chair
Associate Professor, Dept. of Computing Security
Rochester Institute of Technology

Damira Pon, Proceedings Chair
Senior Research Analyst, NYS Center for Information Forensics and Assurance
University at Albany, State University of New York

Karen Sorady, Organization Chair
Assistant Deputy Director of Cyber Programs
Office of Cyber Security (OCS)

Advisory Committee

Gurpreet Dhillon, VCU
Nasir Memon, NYU Polytechnic
Raj Sharman, UBuffalo, SUNY
Bhavani Thuraisingham, UT Dallas
Shambhu Upadhyaya, UBuffalo, SUNY

Technical Program Committee

Alex Tuzhilin, New York University
Anil B. Somayaji, Carleton University, Canada
Ben Shao, Arizona State University
Bill Oblitey, Indiana University of Penn.
Boleslaw Szymanski, RPI
Bradley Malin, Vanderbilt University
Corey Schou, Idaho State University
Daniel Rice, Tech. Solutions Experts, Inc.
Daryl Johnson, RIT
Dipankar Dasgupta, University of Memphis
Ernst Bekkering, Northeastern State University
Farookh Salam, UNC at Greensboro
Gaurav Bansal, University of Wisconsin University
George Berg, University at Albany, SUNY
George Markowsky, University of Maine
Hoang Pham, Rutgers University
Hong C. Li, Intel Corporation **Jim Boardman,** Alfred State Univeristy
Jim Hoag, Champlain University
Jingguo Wang, UT at Arlington
Junjie Wu, Beihang University
Kevin Williams, University at Albany, SUNY
Leon Reznik, RIT
M.P. Gupta, Indian Institute of Tech., Delhi
Manish Gupta, M&T Bank
Martin Loeb, University of Maryland
Merrill Warkentin, Mississippi State
Mohan Kankanhalli, National U. of Singapore
Murtuza Jadliwala, EPFL, Lausanne
Nan Zhang, George Washington University
Peter Stephenson, Norwich University
Raghu Santanam, Arizona State University
Rick Mislan, RIT
S. S. Ravi, University at Albany, SUNY
Shiu-Kai Chin, Syracuse University
Stephen F. Bush, GE Global Research Center
Sumita Mishra, RIT
Tejaswani Herath, Brock University
Thu D. Nguyen, Rutgers University
Tin Kam Ho, Alcatel-Lucent Bell Laboratories
W. Art Chaovalitwongse, Univ. of Washington
Wade Trappe, Rutgers University

8th Annual Symposium on Information Assurance (ASIA '13)

General ASIA Chair:

Sanjay Goel

Information Technology Management, School of Business

University at Albany, State University of New York

Academic Track of 16th Annual NYS Cyber Security Conference

Empire State Plaza Albany, NY, USA

June 4-5, 2013

MESSAGE FROM ASIA GENERAL CHAIR

Welcome to the 8th Annual Symposium on Information Assurance (ASIA'13)! This event complements the NYS Cyber Security Conference as its academic track with a goal of increasing interaction among practitioners and researchers to foster infusion of academic research into practice. For the last several years, ASIA has been a great success with excellent papers and participation from academia, industry, and government and well-attended sessions. This year, we again have an excellent set of papers, invited talks, and keynote address. The keynote speaker for ASIA this year is Billy Rios who is currently the Director of Consulting at Cylance, Inc. and the Chair of the Operational Security Testing panel at the National Board of Information Security Examiners (NBISE). He is a fantastic speaker with a wealth of security experience from Microsoft, Google, and Defense Information Systems Agency. We have been able to attract some of the top information systems researchers in the world to the conference.

I would like to thank the talented technical program committee that has supported the review process for ASIA. In most cases, the papers were assigned to at least three reviewers who were either members of the program committee or experts outside the committee. It was ensured that there was no conflict of interest. The papers and the reviews were also personally read concurring with reviewer assessment. Our goal is to keep the quality of submissions high as the symposium matures. There were multiple levels of quality control – first with the reviewers, then with the program chair, and then with the general chair. The committee serves a critical role in the success of the symposium and we are thankful for the participation of each member.

I am grateful to the advisory committee members for their useful suggestions in managing the conference and would like to thank our ASIA chair, H. Raghav Rao, and Program Chair, Tae (Tom) Oh, for setting up an excellent program for ASIA. The efforts of our review co-chairs, William Stackpole and Sylvia Perez-Hardy, are acknowledged for efficiently managing the review process and coordinated with the authors. Finally, I would like to thank our proceedings chair, Damira Pon, who has worked dedicatedly to review the articles and producing the proceedings with very high quality standards. Thanks also my other doctoral students, Ersin Dincelli and Ethan Sprissler, for their assistance with the review and editing of the papers to support the production of the proceedings.

We were fortunate to have extremely dedicated partners in the Office of Cyber Security (OCS); the NYS Forum; and the University at Albany, State University of New York (UAlbany). Our partners have managed the logistics for the conference, allowing us to focus on the program management. We would like to thank the conference terabyte sponsor AT&T, the conference megabyte sponsors Symantec and Cisco, the symposium dinner sponsor – the University at Albany's School of Business and the conference kilobyte sponsor Northrop Grumman for providing financial support for the symposium.

I hope that you enjoy the symposium and continue to participate in the future. In each of the subsequent years, we plan to have different themes in security-related areas. We also plan to hold the symposium next year. If you would like to propose a track or partner on a future symposium, please let us know. The call for papers for next year's symposium will be distributed in the fall and we hope to see you again in 2014.

Sanjay Goel, ASIA General Chair
Director of Research, CIFA
Associate Professor, School of Business

8th ANNUAL SYMPOSIUM ON INFORMATION ASSURANCE (ASIA '13)

DAY 1: Tuesday, June 4, 2013 (8:00am – 5:00pm)

REGISTRATION & VISIT EXHIBITORS – Base of the Egg (8:00 – 9:00am)

MORNING SESSION & KEYNOTE (9:00 – 10:30am)

Welcome Address: Fran Reiter, *Executive Deputy Director of State Operations*

Keynote: Recommended Cyber Actions for Large Enterprises: An Industry Perspective
Dr. Michael Papay, Northrop Grumman Information Systems

VISIT EXHIBITORS (10:30 – 11:00am) / AT&T DEMO (10:35 – 10:55am)

SYMPOSIUM SESSION 1: Behavioral Security I (11:00 – 11:50am)

Chair: Raj Sharman, *University at Buffalo, SUNY, NY*

Paper: What Drives Perceptions of Threats to Your Facebook Friends' Information?
Stéphane E. Collignon, Tabitha L. James, Merrill Warkentin* and Byung Cho Kim+, *Virginia Tech, VA, Mississippi State University*, MS and Korea University, South Korea+*

Paper: Impact of Security and Privacy Concerns among Medicare Patients on Sharing Health Information Online
Wencui Han, Rohit Valecha, and Raj Sharman, *University at Buffalo, SUNY, NY*

LUNCH ON YOUR OWN / VISIT EXHIBITORS (11:50am – 1:00pm)

SYMPOSIUM SESSION 2: Behavioral Security II (1:00 – 1:50pm)

Chair: Bill Stackpole, *Rochester Institute of Technology, NY*

Paper: Customized Behavioral Normalcy Profiles for Critical Infrastructure Protection
Andrey Dolgikh, Zachary Birnbaum and Victor Skormin, *Binghamton University, NY*

Paper: Consumer Adoption of Smart Metering Technology
Merrill Warkentin, Sanjay Goel* and Philip Menard, *Mississippi State University, MS and University of Albany*, SUNY, NY*

VISIT EXHIBITORS / SYMANTEC DEMO (1:55 – 2:05pm)

SYMPOSIUM SESSION 3: Security Research (2:10 – 3:00pm)

Chair: Sanjay Goel, *University at Albany, NY*

Invited Talk: Challenges in Security / Security Research
H. Raghav Rao, *University at Buffalo, SUNY, NY*

Paper: Security Analysis of Certified Wireless Universal Serial Bus Protocol
Rishabh Dudheria and Wade Trappe, *Rutgers University, NJ*

VISIT EXHIBITORS (3:00 – 3:20pm) / CISCO DEMO (3:00 – 3:15pm)

SYMPOSIUM SESSION 4: Handheld and Wireless Device Security (3:20 – 4:15pm)

Chair: Anil Somayaji, *Carleton University, Canada*

Paper: Malware Analysis for Android Operating System
Kriti Sharma, Trushank Dand, Tae Oh and William Stackpole, *Rochester Institute of Technology, NY*

Paper: Android Malware Analysis Platform
Ben Andrews, William Stackpole and Tae Oh, *Rochester Institute of Technology, NY*

END OF DAY 1

8th ANNUAL SYMPOSIUM ON INFORMATION ASSURANCE (ASIA '13)

DAY 2: Wednesday, June 5, 2013 (8:00am – 4:30pm)

REGISTRATION / VISIT EXHIBITORS – Base of Egg (8:00 – 9:30am)

MORNING SESSION & KEYNOTE (8:30 – 10:00am)

ASIA Welcome Address: Sanjay Goel, *ASIA General Chair*
School of Business, University at Albany, SUNY
Brief Remarks: Jim Dias, *Vice President for Research, University at Albany, SUNY*
ASIA Keynote: Why every CSO needs to know Industrial Control Systems (ICS)
Billy Rios, *Cylance, Inc.*
ASIA Best Paper Award Presentation: Tae Oh, *ASIA Program Chair*
Computing Security Department, Rochester Institute of Technology

VISIT THE EXHIBITORS (10:00 – 10:30am) / AT&T DEMO (10:05 – 10:25am)

SYMPOSIUM SESSION 5: Network Security (10:30 – 11:20am)
Chair: Tae Oh, *Rochester Institute of Technology, NY*

Paper: Detecting Infection Source and Building Predictive Blacklists with an Attack-Source Scoring System Liyun Li and Nasir Memon, *NYU Polytechnic, NY*
Paper: A Private Packet Filtering Language for Cyber Defense
Michael Oehler, Dhananjay S. Phatak and Alan T. Sherman *University of Maryland Baltimore County, MD*

VISIT THE EXHIBITORS (11:20 – 11:40am) / CISCO DEMO (11:25 – 11:35am)

SYMPOSIUM SESSION 6: Data Storage (11:40 – 12:30pm)
Chair: Merrill Warkentin, *Mississippi State University, MS*

Paper: Cloud Security: Attacks and Current Defenses
Gehana Booth, Andrew Soknacki and Anil Somayaji, *Carleton University, Canada*
Presentation: Data Breach Reporting Preparation: An Analysis of Practice
Ernst Bekkering, *Northeastern State University, OK*

LUNCH ON YOUR OWN / VISIT THE EXHIBITORS (12:30 – 1:40pm)

SYMPOSIUM SESSION 7: Education (1:40 – 2:30pm)
Chair: George Berg, *University at Albany, SUNY, NY*

Paper: Mobile Security and Vulnerability Exploitation as a Flipped Classroom Security Curriculum. Richard P. Mislan and Tae Oh, *Rochester Institute of Technology, NY*
Paper: Teaching Android Malware Behaviors for Android Platform using Interactive Labs. Colin Szost, Kriti Sharma, Tae Oh, William Stackpole and Richard P. Mislan, *Rochester Institute of Technology, NY*

NETWORKING BREAK (2:30 – 2:50pm)

SYMPOSIUM SESSION 8: Data Storage, Forensics, and Security (2:50 – 3:45pm)
Chair: Fabio R. Auffant II, *University at Albany, SUNY, NY*

Paper: Discovering Predictive Event Sequences in Criminal Careers. Carl A. Janzen, Amit V. Deokar* and Omar F. El-Gayar*, *University of the Fraser Valley, Canada and Dakota State University, SD**
Paper: A Conceptual Investigation: Towards an Integrative Perspective of Risks in Information Systems Development & Usage. Jim Samuel, *Baruch College - City University of New York, NY*

CLOSING REMARKS (3:45 – 4:00pm)
Sanjay Goel, *Symposium Chair*

TABLE OF CONTENTS

TABLE OF CONTENTS, CONT'D.

What Drives Perceptions of Threats to Your Facebook Friends' Information?

Stéphane E. Collignon, Tabitha L. James, Merrill Warkentin and Byung Cho Kim

***Abstract*— Concerns about privacy in the online environment are quite prevalent and continue to increase with the use of the Internet for both commercial and social activities. Research concerning online privacy has rightfully focused on e-commerce applications and, therefore, on companies' collection and usage of personal information. Therefore, the assumed threat of misusing information has traditionally been viewed as coming from a company. With the introduction of social media, we are increasingly using the Internet for socialization. In this case, we have transferred concerns of improper use of information in social situations from a face-to-face environment to an electronic one. The threats to unwanted disclosure are increasingly coming from third party use of our information. In this paper, we examine how we perceive Facebook use as constituting a threat to others and drivers of this perception. We survey 403 students at a major public US university who are Facebook users. We find that self-efficacy and personal privacy concerns increase one's perception of the severity of the threat to other people caused by the disclosure of one's personal information as a result of online activity, while frequency of use tends to decrease that same perception. We also find that personal privacy concern increases one's perception of the susceptibility of such threats and that the perception of susceptibility increases the threat severity perception.**

***Index Terms*—Facebook, Privacy, Social Networks, Threat Perception**

I. Introduction

FOUNDED IN 2004, Facebook now counts more than 1 billion active monthly users and 618 million active daily users as of December 2012 (http://newsroom.fb.com/Key-Facts). Considering a world population of approximately 7 billion (http://en.wikipedia.org/wiki/World_population), the popularity of Facebook cannot be denied. With an open profile, this constitutes a very large possible audience for Facebook users. There have been a number of privacy concerns with regard to Facebook use. For example, Debatin et al. [1] list the following: "inadvertent disclosure of personal information, damaged reputation due to rumors and gossip, unwanted contact and harassment or stalking, surveillance-like structures due to backtracking functions, use of personal data by third-parties, and hacking and identity theft."

Studies have found that users report being aware of privacy issues stemming from the use of Facebook [1]. Even so, it has been shown in previous research that a substantial amount of users are not aware of (or do not use) the privacy settings afforded to them by social networking sites [2]. It is also the case that even if users appear to understand the privacy risks associated with the use of such sites, it often does not inhibit their actual use [1], [2], [3]. Debatin et al.'s [1] study also found that users were more likely to think that other people were more at risk for negative consequences than themselves.

Previous research has explored the privacy behaviors of individuals with regard to disclosing their own information online. However, as activity on social media sites increase, it is not just one's own information that is disclosed. For example, consider that early users of Facebook would often fill out a profile with information concerning their interests and background and put up a profile picture. Facebook users today tag themselves, along with all of their friends, at a restaurant in real-time; pictures of the outing uploaded from a mobile phone can also be tagged with location and participants. Not only does this type of activity give away the user's location, preference, and image information, but also that of every friend they tagged at the event. Therefore, there is increasing risk to people associated with that Facebook user. The motivation for the current study is to explore Facebook users' perception of the threat to others associated with their own activity on the site.

As use of social media continues to rise and the variety of ways in which to interact through the medium increases, the influence of an individual's use on others will become increasingly important to maintaining everyone's security and privacy. Our study explores individuals' awareness on how their online security behavior can protect others. To this end, we explore individuals' perception of the threat their Facebook activity presents to other people and the drivers of this perception.

The protection motivation theory (PMT) identifies two cognitive mediating processes that combine to form the protection motivation. In the current study, we will borrow the concept of "perception of threat" from one of the cognitive mediating processes, the threat appraisal. PMT suggests that consideration of the severity of the threat and susceptibility or likelihood of the threat occurring forms an individual's perception of threat. Using these two concepts, we explore the

S. E. Collignon is with the Business Information Technology Department, Virginia Tech, Blacksburg, VA 24060 USA. (e-mail: stephane@vt.edu).

T. L. James is with the Business Information Technology Department, Virginia Tech, Blacksburg, VA 24060 USA. (e-mail: tajames@vt.edu).

M. Warkentin is with the Management and Information Systems Department, Mississippi State University, Starkville, MS 39762 USA. (e-mail: m.warkentin@msstate.edu).

Byung Cho Kim is with the Department of LSOM, Korea University, Seoul, Korea. (e-mail: bkim@korea.ac.kr).

perception of the threat to others posed by a user's Facebook activities. Furthermore, we look at factors influencing an individual's perception of severity and susceptibility of disclosing others' personal information caused by their actions. Specifically, we examine frequency of Facebook use, self-efficacy, and privacy concerns as antecedents. All three of these factors have been linked to attitudes related to social network sites [2], [4], [5].

The reminder of the paper will be structured as follows: section 2 presents the theoretical foundation for our model and presents our hypotheses. Then, the methodology and analysis is presented in section 3. Section 4 concludes our paper by discussing the implications of our research.

II. Theoretical Foundation and Hypotheses Development

In *The Dynamics of Group Behaviors*, Elton T. Reeves explains that "there must be a collective value judgment that the threat exists" [6] for a group to act upon a threat. This idea motivates the current study to explore if people perceive a threat to others when they disclose information on Facebook containing information about other people. Additionally, if the perception of a threat exists, the factors influencing that perception are useful to explore. If drivers to the threat appraisal can be determined, techniques to motivate protection behaviors of others can be created and assessed. In the social media realm, it can be argued that the threats to both oneself and to others have not been fully assessed, which makes the exploration of peoples' perception of the domain very timely. While some literature has suggested threats, with regard to privacy, to oneself of data disclosure and collection activities [7], [8] and social network use [1], [2], [9], there is no literature regarding the threat to others of one's social network use, though perceptions of threats to others are found in other contexts.

Research has shown that people can perceive threats to others and also that this perception can be provoked through messages increasing the emotional link to others [10]. Shelton and Rogers [10] suggest that "fear appeals can persuade us to protect ourselves, other people, and even infrahuman animals." The study, centered on pro-environmental action organizations, reports that the noxiousness of the threat and the efficacy of the mechanism to cope with the threat are crucial to motivating people to help others [10]. This study is one of the few that approach the matter of threats to others, using concepts from PMT, where the self is not considered part of the others.

Given this background, we adopt dependent variables based on the traditional constructs of the threat appraisal, threat susceptibility and threat severity, created by Rogers [11] within the framework of the PMT. PMT has been successfully applied to information systems (IS) research, most markedly in the information security area [12], [13]. It is useful to adapt the definitions of the threat appraisal constructs to our domain. Threat susceptibility (TSUS), in our study, is the Facebook user's perception of the likelihood of others experiencing the threat of having personal information disclosed because of the user's behavior. Similarly, threat severity (TSEV) is a Facebook user's perception of the seriousness of the threat of exposing others' personal information through his or her Facebook activities. We define personal information as: any information about the person in question, whether textual (for example: a person's phone number, email address, twitter handle, birthday, etc.; a statement about where a person is physically located at a given moment; a statement about a person's opinion on a television show, movie, book, politician, etc.; a statement about what a person is wearing or looks like; information about a person's relationship or family relations; etc.) or graphical (a picture of a person, a video or a person). In order to measure these constructs, we adapted scales validated in a previous IS study to our context [12].

In this study, we examine three antecedents to threat susceptibility and severity: frequency of Facebook use (USE), self-efficacy (SEFF) - user's perception of their ability to competently use Facebook, and privacy concern (PRIV) - the user's concern for their own privacy related to Facebook usage. Research has indicated that students with a high level of activity on Facebook were more likely to have a private profile [5]. This could indicate that the more users use Facebook, the more of their own information they provide to the site which inclines them towards the use of some controls to protect that personal information. However, research has also demonstrated a lack of willingness of Facebook users to use privacy controls [9], unless users have actually experienced a privacy invasion due to information leakage on the site [2]. This indicates that users may have difficulty in attaching consequences to their online behaviors on Facebook. There is also evidence that other desires, such as a need for popularity, are important drivers of disclosure [3], which could override privacy concerns. In fact, the idea that some desires overwhelm an individual's need for privacy, referred to as the privacy calculus, comes from the social sciences [14] and has been applied to IS e-commerce situations [15]. We argue that frequency of use provides a comfort level with the platform that, especially in combination with positive rewards (socialization), could lull users into a false sense of security. We suggest that the combination of difficulty in attaching consequences to Facebook activities, in conjunction with a fulfilled socialization desire and an increased comfort level, causes individuals with a high frequency of use to perceive the threat of their Facebook activity to others as minimal (both in terms of likelihood of a negative privacy experience and severity of an experienced event). Similarly, a user who does not frequently use Facebook should perceive a higher threat, because the unknown should appear riskier. This leads us to our first proposition:

P1: Frequency of use will impact perceived susceptibility and severity of threat to others from the user's Facebook activity.

Another argument supporting low frequency of use and high perception of threat is that lack of use could be seen as a coping mechanism. That is, one way to cope with a threat is to avoid it [16], [17]. In *L'éloge de la fuite* from Henri Laborit [16], it is suggested that the possibility and the severity of a problem should increase avoidance. Therefore, if a user has experienced a past information leak, they may perceive an

increased likelihood and severity of the threat and, as a consequence, be less frequent users. Some support for this has been provided, in that, users have been demonstrated to become more careful users after a privacy experience [2]. This indicates a feedback loop that develops privacy concern over time as an effect of experience. This process is suggested in classic privacy theory [14]. We used a scale to measure frequency of Facebook use, based on the items used by [18].

Frequency of use captures a user's exposure to the platform (Facebook), but it does not depict the usage ability of the user. More adept Facebook users may have more possibilities of exposing another person's information since they typically have a wider variety of activities available to them. Therefore, a proficient Facebook user may have a different perception of what damages other people could be exposed to via his or her Facebook activity. Thus, following the recommendation of Burton-Jones and Straub [19], we decided to extend our consideration of usage beyond frequency of use. In order to explore proficiency, we adopted the construct of self-efficacy.

We define self-efficacy as the respondent's self-reported ability to use Facebook. We follow Bandura [20] in that self-efficacy is based on "people's judgments of their capabilities to […] execute courses of action […]" and "it is concerned not with the skills one has but with judgments of what one can do with whatever skills one possesses" [20, pp. 391]. Computer self-efficacy has been widely studied in the IS literature [21]. Forms of Internet self-efficacy have also been proposed [4], [22]. Gangadharbatla [4] examined the relationship between Internet self-efficacy and the attitude towards social networking sites and found that self-efficacy had a positive impact on attitude. Although, the current study is borrowing from PMT's threat appraisal cognitive mediating process, we do not use this theory with regard to self-efficacy. In PMT, self-efficacy is defined as the perceived ability to implement the adaptive response to the threat [23], which we are not measuring in our study. In short, we are borrowing from PMT a small concept – the threat appraisal, and not testing the full theory. Our self-efficacy construct is similar to the computer and Internet self-efficacy constructs mentioned above. Similar to computer self-efficacy studies, we use Facebook self-efficacy as an antecedent to a perception (since we are not measuring behaviors in this study). We argue that more proficient Facebook users (those with higher self-efficacy) will perceive a higher threat susceptibility and severity to others from his or her Facebook activity. We suggest that this is due to the availability of more skills, and thus, more possibilities of exposing others' information. We also submit that as the level of use advances, Facebook users will become more aware of the privacy risks to themselves and others. This leads us to the following proposition:

P2: Perceived self-efficacy will impact the perceived susceptibility and severity of threat to others from the user's Facebook activity.

The measurement scale for Facebook self-efficacy in this context is based on one used by Bélanger et al. [24]. We also followed the suggestion of Burton-Jones and Straub [19] and identified tasks that are of importance to us in the usage of Facebook. Based on the consultation of a panel of experts and the findings of the social media association in the college of business at Virginia Tech [25], we constructed our items around the perceived ability to post pictures, status changes, and comments on somebody's wall and to tag people in Facebook. Doing so, we address two preoccupations at the same time. First, with regards to Marakas et al. [26], we want to measure the perceived ability of the users to use Facebook at different levels of difficulty. Second, we want to consider actions that could involve others.

The last antecedent we consider in the model (Fig. 1) is privacy concern. With this construct, we are focused on the Facebook user's concern for his or her own privacy. We suggest that an individual's own concern should translate to concern vis-à-vis others. Privacy in today's data-driven world has become a very focal topic. For comprehensive overviews of the state of privacy research in the IS discipline, we will refer the interested reader to Smith et al. [27] and Bélanger and Crossler [28]. For the present study, we will adopt a definition of individual privacy concern similar to that of Dinev and Hart [29]: concerns of a user regarding possible loss of that user's privacy as a result of voluntary information disclosure to Facebook. We used a contextually modified scale to measure privacy concern based on the one given in Dinev and Hart [29]. We postulate that a user with high individual privacy concerns related to his Facebook activities will view the threat to others as similarly high. In other words, we suggest transference of concern from the individual to others. This leads to the following proposition:

P3: Privacy concern will impact perceived susceptibility and severity of threat to others from the user's Facebook activity.

PMT suggests that the perception of threat is formed by the perceived severity and susceptibility of the threat. Risk management standards in information security calculate risk as a function of the likelihood and severity of a threat [30]. The theory of deterrence from criminology suggests that criminal sanctions vary in effectiveness due, in part, to the perceived severity and certainty of the punishment [31]. In each case, although cast in different contexts, susceptibility and severity are related. We model these constructs as being related and suggest that if a user believes others are highly susceptible to being threatened by his or her Facebook activity; he or she will perceive the threat to others to be more severe.

P4: Perceived susceptibility of threat to others from the user's Facebook activity will impact the perception of severity of threat to others from the user's Facebook activity.

III. Methodology and Analysis

A. Scale Development and Survey Administration

The scales for the current study were presented to an expert panel consisting of faculty members and Ph.D. students from Information Systems and Marketing departments who are experts on survey development, as well as undergraduate and master's students in marketing and information systems that

are frequent Facebook/social media users that were used as context experts. Comments on the phrasing and appropriateness of the scale items were solicited from the expert panel. The feedback was used to reassess and adapt items.

The survey was pilot tested twice. The first pilot was conducted in the summer of 2012. Data for the first pilot was collected from two major universities in the United States (U.S.). The sample size for this pilot was 76. After a couple of minor reformulations of items, a second pilot was administered during the fall of 2012 with a sample size of 92. The exploratory factor analysis (EFA) for the second pilot revealed an acceptable instrument. A full data collection was conducted in the fall of 2012 at one large southeastern U.S. university. All surveys were administered in an online format using Qualtrics[1] and participation was voluntary.

The number of respondents for the final data collection is 440. The survey questions were randomized and a manipulation check was included to improve data quality. Our response set needs to include only Facebook users. To that end, respondents were asked if they had logged into their Facebook account several times in the last month. If they did not meet the criterion, they were excluded from the sample. This excluded only seven participants from the study. Responses that failed an embedded manipulation check were also removed, which resulted in 25 responses eliminated from the final sample. Incomplete responses were excluded from the final sample, yielding a final sample size of 403 responses.

Demographic information was collected on the survey participants and is given in Table I. The number of males is slightly higher than the number of females. Most of the participants range in age from 18 to 25. This was to be expected due to the venue that the data was collected, but is also appropriate considering the need for the respondents to be frequent Facebook users. A majority of the sample is Caucasian/White and full-time students.

In addition to the demographic information, technical proficiency data and Facebook statistics were also collected. This data is summarized in Table II. This data is self-reported but provides an overview of the respondents' comfort with technology and social media interest. Most of the sample reported a technical proficiency of an intermediate to advanced level. The majority of the sample has possessed a Facebook account for more than 2 years and has over 300 Facebook friends. This indicates the respondents believe themselves to be savvy computer users that have used Facebook for several years and have a large number of connections.

The descriptive statistics for the survey items are shown in Table III. An examination of the means illustrates some interesting relationships. Overall, the respondents feel that the consequences of someone's personal information being leaked as a result of his Facebook activity would be severe (mean approximately "agree" for all four items). However, the survey respondents do not appear to believe that others are very susceptible to their information being disclosed due to the respondents' actions Facebook (mean approximately "neutral" for all three items). The standard deviations are larger for the susceptibility items than the severity items, indicating more variation in the opinions over susceptibility.

TABLE I
SAMPLE DEMOGRAPHIC INFORMATION

Gender		Age		Ethnicity	
Male	218	18 to 20	261	Caucasian/White	332
Female	*185*	21-25	131	African American/Black	51
		26-30	3	Latino/Hispanic (white, black)	7
		31-35	3	Pacific Islander	0
		36-54	4	Native American/Indian	1
		55 or older	0	Middle-Eastern	1
				Mixed Race	4
				Other	1
				Asian	5

Employment Status	
Employed full-time	3
Full-time student	348
Employed part time or looking for full-time work	2
Work part time and go to school part time	26
Unemployed, not looking for work	8
Other	16

TABLE II
TECHNICAL EXPOSURE OF SAMPLE

TECHNICAL PROFICIENCY		Length of Time on Facebook	
Novice	11	Less than 6 months	1
Intermediate	242	6 months to 1 year	2
Advanced	150	1 to 2 years	12
		2 to 4 years	101
		4 or more years	287
		55 or older	

Number of Friends on Facebook	
1 to 30	2
31 to 100	7
101 to 300	46
301 to 500	95
501 to 1000	130
1001+	123

The means for the use items are a little lower than might be expected (between "neutral" and "agree"). The items with the highest mean and the lowest standard deviation refer only to "checking" Facebook regularly, while the other two point towards use. This could indicate that our sample, on average, uses Facebook as an informational entertainment source more than they actively use the more advanced interactivity features. Regarding self-efficacy, although, they feel very comfortable with using most basic features of Facebook that would let them interact with others by sharing information. Considering their own privacy concerns related to Facebook, the means were between "Neutral" and "Agree." This is indicative of some concerns relative to the respondents' own privacy with the use of Facebook.

[1] http://www.qualitrics.com

TABLE III
STATISTICS FOR SURVEY ITEMS 5-POINT LIKERT SCALE (1 = STRONGLY DISAGREE TO 5 = STRONGLY AGREE)

Construct Indicator	Item	Mean	Standard Deviation
TSEV1	The posting of somebody's personal information resulting from my Facebook activities could have severe consequences for that other person.	3.79	.863
TSEV2	If I released somebody's personal information through Facebook, it could be harmful for that other person.	4.07	.781
TSEV3	If another person's personal information was exposed by my use of Facebook, this could be significant for that person.	3.92	.811
TSEV4	It could be unfortunate for a person if his or her personal information was spread by my Facebook activity.	4.03	.834
TSUS1	It is possible that other people's personal information may be released by my use of Facebook.	3.34	1.086
TSUS2	If I use Facebook, it is likely that the personal information of some other people may be posted.	3.10	1.100
TSUS3	Others may experience leaks of personal information because of what I do on Facebook.	2.80	1.064
USE1	I use Facebook frequently.	3.84	1.001
USE2	I check my Facebook page frequently.	3.91	.954
USE3	Facebook is part of my daily activity.	3.61	1.155
SEFF1	I know how to post pictures on Facebook.	4.35	.736
SEFF2	I know how to update my status.	4.46	.615
SEFF3	I know how to tag people on Facebook.	4.38	.703
SEFF4	I know how to post something to a friend's wall.	4.42	.676
PRIV1	I am concerned because information I transmit on Facebook can be intercepted by third parties.	3.46	.890
PRIV2	I am concerned about submitting personal information on Facebook because of what others might do with it.	3.60	.953

B. *Statistical Results*

The factor analysis is shown in Table IV. For a sample size of n=403, the factor loadings are all more than adequate. All of the items in our instrument load highly on the appropriate factor. There were no significant cross-loadings (>0.2, all loadings <0.2 were suppressed in Table IV). Table V shows the Cronbach's Alphas for each factor. All factors have a Cronbach's Alpha of > 0.7, which means our scale is reliable.

TABLE IV
FACTOR ANALYSIS (PRINCIPAL AXIS FACTORING, PROMAX ROTATION)

	Factor				
	1	2	3	4	5
TSEV1			.601		
TSEV2			.814		
TSEV3			.775		
TSEV4			.811		
TSUS1					.765
TSUS2					.914
TSUS3					.661
USE1		.877			
USE2		.896			
USE3		.904			
SEFF1	.813				
SEFF2	.893				
SEFF3	.950				
SEFF4	.930				
PRIV1				.702	
PRIV2				.903	
PRIV3				.897	

The model fit statistics are given in Table VI. Using only the χ^2, our model indicates inadequate fit since $p < 0.05$. However, this manner of judging the model fit is known to be sensitive to large sample sizes [32]. It is common in the literature to look at other fit statistics to gauge model fit that adjust for sample size. It is suggested that the SRMR should be < 0.08 and the RMSEA < 0.06 and the relative fit indices (IFI, TLI, NFI) be > 0.95 to suggest adequate model fit [32], [33]. For the model under consideration in the current study, all of the relative fit statistics are at or above 0.95. In addition, the SRMR is 0.04 and the RMSEA is 0.05. All of these fit statistics indicate an adequate fit.

TABLE V
RELIABILITY

Factor	Cronbach's Alpha
TSEV	.941
USE	.919
SEFF	.838
PRIV	.871
TSUS	.822

The final model is shown in Fig. 1, which shows the standardized estimates associated with all proposed relationships. A significant relationship was not found between the frequency of Facebook usage and the susceptibility of others' information to release through the respondents' Facebook activities. All other relationships were significant.

TABLE VI
MODEL FIT STATISTICS

Good of fit measures	χ^2 /df	CFI	IFI	TLI	NFI	GFI	AGFI	SRMR	RMSEA	PCLOSE
Good model fit	<3.00	>0.90	>0.95	>0.95	>0.95	>0.95	>0.80	<0.09	.05-.10	>.05
CFA Model	2.168	0.972	0.972	0.965	0.949	0.935	0.909	0.040	0.054	0.239
SEM Model	2.151	0.972	0.972	0.965	0.949	0.935	0.909	0.041	0.054	0.259

IV. IMPLICATIONS AND CONCLUSIONS

This study contributes to the literature by being the first to explore users' perceptions of the threat to others' personal information by their Facebook activities. Overall, our study confirms that people's perception of threat to others exists and varies in accordance with some antecedents. We find that frequency of use is negatively related to perception of threat severity to others from a user's Facebook activity. However, the relationship between frequency of use and perception of threat susceptibility to others was not significant. This suggests that as people are exposed more often to Facebook, either the rewards obtained by that use overshadow the perceived severity of the threats or the comfort level with the platform increases, lulling the user into a false sense of security that is transferred to their perceptions of threat severity to others. The users may rationalize that they are obtaining so much enjoyment from the site, that anything that goes wrong couldn't be that serious. Considering the lower means for susceptibility, the users already feel that it is not highly likely that there is a threat to others from their Facebook use. This could be argued to reinforce the idea that

it is hard to imagine the consequences of information disclosure until one has experienced a privacy event.

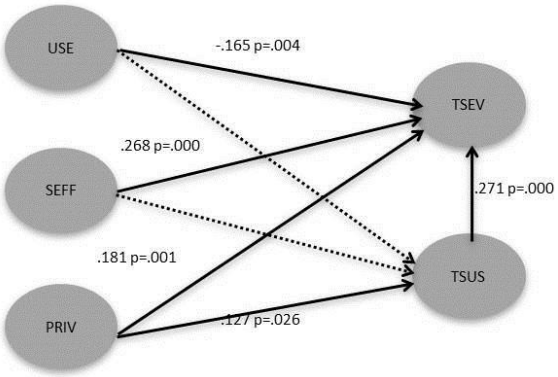


Fig. 1. Model for perceived threat to others' by Facebook use.

As hypothesized, as a user's self-efficacy increases, so does the perception of severity to others of his or her use. Therefore, as the user's proficiency increases, so does his or her perception of threat severity to others, but not susceptibility. Similar logic to frequency of use could apply for the non-significant relationship from self-efficacy to the perception of threat susceptibility. As a user's perceived proficiency increases, so does his or her perception of the severity of threat to others, which indicates that awareness or use training could perhaps mitigate risky behaviors.

An individual's own privacy concern impacts both perceptions of threat severity and threat susceptibility to others. This lends support for transference of concern to others. As individual privacy concerns related to Facebook increase, so does the perception of threat severity and susceptibility to others. This is promising in that if misconceptions of risk towards Facebook use can be corrected, others' privacy would also benefit. Perception of threat susceptibility to others is shown to impact the perception of threat severity to others. This suggests that the less likely Facebook users perceive a threat to others to be, the less severe they will perceive the threat to be. This result also points to the benefit of awareness training and to users being able to attach consequences to their use a little more easily.

REFERENCES

[1] B. Debatin, J.P. Lovejoy, A.K., Horn and B.N. Hughes, "Facebook and online privacy: attitudes, behaviors, and unintended consequences," *J. of Comput.-Mediat. Commun.*, vol. *15,* no. *1,* pp. *83-108,* 2009.

[2] A. Acquisti and R. Gross, "Imagined communities: awareness, information sharing, and privacy on the Facebook," in *Privacy Enhancing Technologies,* Berlin/Heidelberg: Springer, 2006, pp. 36–58.

[3] E. Christofides, A. Muise and S. Desmarais, "Information disclosure and control on Facebook: are they two sides of the same coin or two different processes?," *CyberPsychology & Behavior*, vol. *12,* no. *3,* pp. *341-345,* 2009.

[4] H. Gangadharbatla, "Facebook me: collective self-esteem, need to belong, and internet self-efficacy as predictors of the iGeneration's attitudes toward social networking sites," *J. of Interactive Advertising*, vol. *8,* no. *2,* pp. *5-15,* 2008.

[5] K. Lewis, J. Kaufman and N. Christakis, "The taste for privacy: an analysis of college student privacy settings in an online social network," *J. of Comput.-Mediat. Commun.*, vol. *14,* no. *1,* pp. *79-100,* 2008.

[6] E.T. Reeves, *The dynamics of group behavior*. American Management Association, 1970, pp. 51.

[7] E.M. Kirsh, D.W. Phillips and D.E. McIntyre, "Recommendations for the evolution of cyberlaw," *J. of Comput.-Mediat. Commun.*, vol. *2,* no. *2,* 1996.

[8] R.O. Mason, "Four ethical issues of the information age, *MIS Quart.*, vol. *10,* no. *1,* pp. *5-12,* 1986.

[9] R. Gross and A. Acquisti, "Information revelation and privacy in online social networks," in *Proc. of the 2005 ACM Workshop on Privacy in the Electronic Society*, 2005, pp. 71-80.

[10] M.L. Shelton and R.W. Rogers, "Fear-arousing and empathy-arousing appeals to help: the pathos of persuasion," *J. Appl. Soc. Psychol.*, vol. *11,* no. *4,* pp. *366-378,* 1981.

[11] R.W. Rogers, "A protection motivation theory of fear appeals and attitude change," *J. Psychol.*, vol. *91,* no. *1,* pp. *93-114,* 1975.

[12] A.C. Johnston and M. Warkentin, "Fear appeals and information security behaviors: an empirical study," *MIS Quart.*, vol. *34,* no. *3,* pp. *549-566,* 2010.

[13] T. Herath and H.R. Rao, "Protection motivation and deterrence: a framework for security policy compliance in organisations," *Eur. J. Inform. Syst.*, vol. *18,* no. *2,* pp. *106-125,* 2009.

[14] R.S. Laufer and M. Wolfe, "Privacy as a concept and a social issue: a multidimensional developmental theory," *J. Soc. Issues*, vol. *33,* no. *3,* pp. *22-42,* 1977.

[15] T. Dinev and P. Hart, "An extended privacy calculus model for e-commerce transactions," *Inform. Syst. Res.*, vol. *17,* no. *1,* pp. *61-80,* 2006.

[16] H. Laborit, *L'éloge de la fuite*. Paris: Laffont, 1974.

[17] R.R. McCrae, "Situational determinants of coping responses: loss, threat, and challenge," *J. Pers. Soc. Psychol.,* vol. *46,* no. *4,* pp. *919-928,* 1984.

[18] C. Ross, E.S. Orr, M. Sisic, J.M Arseneault, M.G. Simmering, and R.R. Orr, "Personality and motivations associated with Facebook use," *Comput. Hum. Behav.*, vol. *25,* no. *2,* pp. *578-586,* 2009.

[19] A. Burton-Jones and D.W. Straub, "Reconceptualizing system usage: an approach and empirical test," *Inform. Syst. Res.,* vol. *17,* no. *3,* pp. *228-246,* 2006.

[20] A. Bandura, "The explanatory and predictive scope of self-efficacy theory." *J. Soc. Clin. Psychol.*, vol. *4,* no. *3,* pp. *359-373,* 1986.

[21] D.R. Compeau and C.A. Higgins, "Computer self-efficacy: development of a measure and initial test," *MIS Quart.*, vol. , no. , pp. *189-211,* 1995.

[22] H.M. Hsu and C.M. Chiu, "Internet self-efficacy and electronic service acceptance," *Decis. Support Syst.*, vol. *38,* no. *3,* pp. *369-381,* 2004.

[23] D.L. Floyd, S. Prentice-Dunn and R.W. Rogers, "A meta-analysis of research on protection motivation theory," *J. Appl. Soc. Psychol.*, vol. *30,* no. *2,* pp. *407-429,* 2000.

[24] F. Bélanger, S.E. Collignon, K. Enget and E. Negangard, "User resistance to mandatory security implementation," in *Proc. of the 2011 Dewald Roode Information Security Workshop*. IFIP WG8.11/11.13, 2011.

[25] PRISM, "Pamplin social media research report presentation," Virginia Tech, 2011.

[26] G.M. Marakas, Y.Y. Mun and R.D. Johnson, "The multilevel and multifaceted character of computer self-efficacy: toward clarification of the construct and an integrative framework for research," *Inform. Syst. Res.*, vol. *9,* no. *2,* pp. *126-163,* 1998.

[27] H.J. Smith, T. Dinev and H. Xu, "Information privacy research: an interdisciplinary review," *MIS Quart.*, vol. *35,* no. *4,* pp. *989-1016,* 2011.

[28] F. Bélanger and R.E. Crossler, "Privacy in the digital age: a review of information privacy research in information systems," *MIS Quart.*, vol. *35,* no. *4,* pp. *1017-1042,* 2011.

[29] T. Dinev and P. Hart, "Internet privacy concerns and social awareness as determinants of intention to transact," *Int. J. Electron. Comm.*, vol. *10,* no. *2,* pp. *7-29,* 2006.

[30] B. Karabacak and I. Sogukpinar, "ISRAM: information security risk analysis method," *Comput. Secur.*, vol. *24,* no. *2,* pp. *147-159,* 2005.

[31] M. Silberman, "Toward a theory of criminal deterrence," *Am. Sociol. Rev.*, vol. *41, June,* pp. *442-461,* 1976.

[32] L. Hu and P. M. Bentler, "Cutoff criteria for fit indexes in covariance structure analysis: conventional criteria versus new alternatives," *Structural Equation Modeling,* vol. *6,* no. *1,* pp. *1-55*, 1999.

[33] B.M. Byrne, *Structural equation modeling with AMOS: Basic concepts, applications and programming*. Mahwah, NJ: Lawrence Erlbaum Associates, 2001.

Impact of Security and Privacy Concerns among Medicare Patients on Sharing Health Information Online

Wencui Han, Rohit Valecha, Raj Sharman

***Abstract*— The use of Internet for sharing health-related information has served to empower patients, improve self-health management knowledge and provide emotional support to patients. This paper investigates the factors that influence the use of Internet to share health-related information by patients covered under Medicare, especially, the impact of security and privacy concern. This paper adapts the Information System Success Model to health care context. The paper proposes that information quality, system security and privacy, physician patient communication, negative emotions and perceived health status are related to share health-related information online. We also examine the impact of education level on health information sharing online. The paper utilizes the 2012 Health Information National Trends Survey (HINTS) to draw conclusions.**

***Index Terms*— Health-related Information, Medicare, Internet use, Security and Privacy**

I. Introduction

THERE IS a growing number of websites and social networks providing health care related information and opportunities for patients to both share information and meet other patients. Examples include Yahoo health, which has over 20 million unique visits every month, and WebMD, which has close to 20 million unique visits every month1. Many studies have indicated the benefit of using Internet to share health-related information. Patients use the information they exchange to self diagnose symptoms [11], understand the treatments and how to use medication devices [9], [17], create or solidify relationships with similar users [10], and better cope with symptoms [23], [34]. However, there are still some issues about using Internet for health care related information preventing patients from utilize this resource. Such as the privacy and security issue [43] the inaccurate patient generated information [1] and so on.

W. Han is with Management Science and Systems Department, State University of New York, Buffalo, NY 14260 USA.
R. Valecha is with Management Science and Systems Department, State University of New York, Buffalo, NY 14260 USA (email: valecha@buffalo.edu).
Raj Sharman is with Management Science and Systems Department, State University of New York, Buffalo, NY 14260 USA (email: rsharman@buffalo.edu).

[1] http://www.ebizmba.com/articles/health-websites

Use of Internet can be complicated for the disabled and elderly populations due to their physical conditions and lack of social media expertise [28]. Medicare is a national social insurance program that guarantees access to health insurance for Americans ages 65 and older and younger people with disabilities as well as people with end stage renal disease. In 2010, Medicare provided health insurance to 40 million Americans. Much of the governmental expense in the United States on health care is attributable to expenses in the Medicare [39]. People covered by Medicare are an important segment of the population that has been understudied in the literature from the Information Systems (IS) perspective.

By adapting the IS Success Model, using the 2012 Health Information National Trends Survey (HINTS) data, we developed and tested a model to study the factors influencing the use of Internet for sharing health-related information by patients covered under Medicare. And we investigate the effect of education.

The remainder of this paper is organized as follows: Section II reviews related literature and Section III highlights our research model. The data and results are summarized in Section IV. The implications, limitations and future research are discussed in Section V.

II. Literature Review

A. Use of Internet for Sharing Health-Related Information

Internet use for health information is growing in popularity with the advent of chat rooms, blogs, discussion boards and online communities [40]. There is a growing body of literature investigate the motivating factors, barriers, channel preference, purposes of patient online information sharing behavior. Prior research suggests that patients use Internet for information sharing to gain psychological support [22], [41], [44], gain knowledge for self-management of disease [15], [12] and to facilitate making treatment decisions [46]. Other benefits include increasing intention to communicate with healthcare providers and a perceived sense of empowerment [30].

B. Medicare Population and Health Information Sharing through Internet

A study by Dickersen et al. [7] reports that patients with Medicare are less likely to search the web for health information compared to the patients with commercial

insurance. Brodie et al. [5] call for online systems to be developed by government for the Medicare population. Gutierrez [13] posits that Medicare providers can improve the healthcare delivery to their providers by making use of the connectivity and the communication infrastructure of the Internet.

C. System Security and Privacy and Use of Internet to Share Health-Related Information

The use of social networking websites for sharing health-related information has prompted concerns about the risks that these websites pose to the security and privacy of personal health information [48]. Bansal et al. [4] point out that individuals' intention to disclose health-related information depends on their trust, privacy concern, and information sensitivity.

D. Information Quality and Use of Internet to Share Health-Related Information

Kisekka and Sharman [20] have pointed out in their literature review that the quality of information exchanged is closely related to the online health-related information sharing behavior. Prior research on the quality of online health information is inconclusive. Some scholars found evidence for the presence of erroneous, incomplete, or misleading information, while others found the quality of information to be acceptable.

III. Model Development

In this paper, we develop a conceptual model by adapting the Information System (IS) Success Model. The IS success model was proposed by DeLone and McLean in 1992 to provide a general and comprehensive definition of IS success that covers different perspectives of evaluating information systems [6]. A large body of literature has adapted IS success model to predict system use in various contexts (please see the meta analysis conducted by Petter and Mclean [31] for more details). The IS success model consists of six dimensions: information and system quality, use, user satisfaction, as well as individual and organization impact. The model suggests that an information system can be evaluated in terms of information and system quality. These characteristics of the system affect the use of the system and user satisfaction. Finally, there will be individual and organizational impact from using the system.

We adapt the IS success model to investigate what factors promote Internet use behavior in the health care context. In addition to information quality and system quality, we propose patients' concerns about their health status and patients' negative emotions as well as physician patient communications affect the use of Internet for sharing health-related information. A conceptual model is developed and presented in Fig. 1.

A. Internet Use

The online health information exists in a variety of formats ranging from traditional web pages to emerging social media, including blogs and social network sites. The users can search for these health-related information, and also share it with patients with similar medical conditions or physicians, etc. Use Internet to share health information is defined as whether the patients have used the Internet to visit a social-networking site or join an online support group to read and share about medical information.

B. Information Quality

Information quality is the desirable characteristics of the system outputs. It has many dimensions such as accuracy, relevancy, and timeliness and so on [42]. In health care context, patients use information found on the Internet to assess and diagnose their diseases as well as decision making process. If relevant information is hard to find or understand, the information provided is not relevant, accurate or trust worthy, patients will shift to other channels to acquire information [8]. Hence the higher information quality perceived by patients, the more likely they will use Internet to search and share information. Thus we hypothesize:

H1: Information Quality is positively related to use Internet to share health information

C. System Quality

System quality is the desirable characteristics of an information system, such as the ease of use, system reliability, flexibility and security [34]. In our study, we focus on the perceived system security and privacy. As in health care context, the privacy and security of information are critical success factors of the system [2], [27].

System Security and Privacy

System security and privacy represent the patient's confidence in safeguarding against security threat on their information and unauthorized access to their information. The Medicare patients feel vulnerable to threats on their health information because of their lower levels of Internet use expertise [16]. Patients are more likely to share health-related information online if they believe they are protected against such security and privacy threats. Thus, we hypothesize:

H2: System Security and Privacy is positively related to use Internet to share health information

D. Physician Patient Communication

The physician patient communication arises out of need to acquire more information from physician [38]. The physician helps the patient understand the severity and the nature of illness, understand treatment options and make decisions about further visits [25], [29]. In case of lack of clear explanations or low quality of service, the patients seek to use other information channels as complements. The Internet provides accessible way to validate the information patients obtained during their encounters with physicians. Thus, we hypothesize:

H3: Physician Patient Communication is negatively related to use Internet to share health information

E. Patients' Emotions and Concerns

Perceived Health Status

Perceived health status refers to one's perception of their overall health. It is an indicator of health status and health risk factors [18], and it is associated with physical factors such as

fitness, illnesses, etc. [37]. Research has shown that specific health events such as being diagnosed with new health problem, having an ongoing medical condition and being prescribed new prescription or treatment leads to online health information search and sharing behavior [35]. The worse the patients' perception about their health, the more likely they feel the need to collect information to understand their condition and evaluate treatments. Thus, we hypothesize:

H4: Perceived health status is negatively related to use Internet to share health information

Negative Emotions

Patients experience stress, hopelessness, and anxiety due to their physical condition. In order to shed their negative emotions they use media like Internet to satisfy their emotional needs by sharing information with people suffering from similar condition, seeking emotional support, and looking for information about the condition they have [45]. Thus we hypothesize:

H5: Feeling negative emotions is positively related to Internet use to search for health information

Level of education is used as the control variable in the model. In summary, we propose the model shown in Fig. 1 to test the hypothesis.

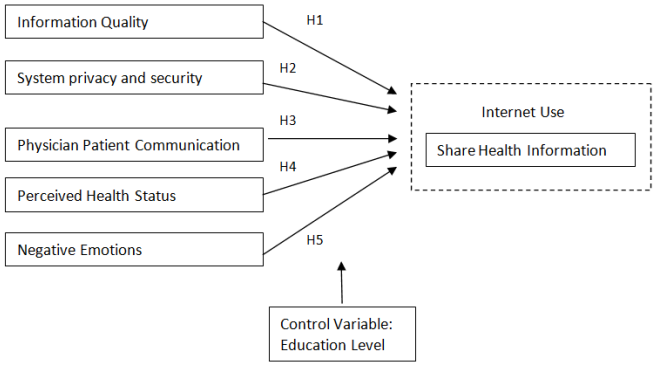


Fig. 1. Research model.

IV. RESEARCH METHODOLOGY

A. Data

This study uses the data collected by 2012 health information national trends survey (HINTS). HINTS is conducted by the National Cancer Institute to study people's health information behaviors. The survey reported 3950 responses, we selected the patients covered under Medicare, and finally 1125 cases were used for analysis. The demographics of the participants are summarized in Table 1:

TABLE I
PARTICIPANTS DEMOGRAPHICS AND DESCRIPTIVE STATISTICS

	Group	Frequency	Percentage
Gender	Male	481	40.7%
	Female	644	54.5%
	Missing	57	4.8%
Education	Less than High School	183	15.5%
	High School Graduate	310	26.2%
	Some College	345	29.2%
	College Graduate or More	310	26.2%
	Missing Data	34	2.95%
Language	Completely and comfortable	1038	87.8%
	Somewhat and not at all comfortable	63	5.8%

B. Instrument Testing Results

We selected the items related to our study from the survey, the constructs and items are summarized in Appendix A.

To test the psychometric properties of the measurement scales, we used SmartPLS software package to test the instrument [36]. The results in Appendix B and C show that we met the statistic requirements for convergent validity, individual item reliability, composite reliability, and discriminate validity [3],[19].

C. Research Model Testing Results

As the data were self-reported and collected through the same questionnaire, there is a potential for common method biases [33]. We performed several tests (Harman's single-factor test, unmeasured latent method construct [ULMC] approach in partial least squares [PLS]) to examine if common method bias is a concern for this study. The results are presented in Appendix D. The results show that common method bias was not a major concern in this study.

The research model was subsequently tested. Use Internet to share health-related information —the dependent variable in this study is a categorical variable with two categories. A binary logistic regression procedure was used to test the model (see the following model). The results are reported in Appendix E. We first examine the correlations among the independent variables. The highest correlation among the independent variables was 0.42, between "Perceived Health Status" and "Negative Emotions", so multicollinearity was not a concern for this analysis [14], [32].

$$\text{Ln}\,[p/(1-p)] = \alpha + \beta 1\ \text{Information Quality} + \beta 2\ \text{System Security and Privacy} + \beta 3\ \text{Physician Patient Communication} + \beta 4\ \text{Negative Emotions} + \beta 5\ \text{Perceived Health Status} + e$$

Then, we examine the impact of level of education. The results are summarized in Appendix F. We use Z-test to investigate the differences between the beta coefficient for two groups. The results show that there is no significant difference between the two groups. Hence education level does not significantly affect the relationships between the factors and Internet use.

V. DISCUSSION

The results of our study show that information quality is positively associated with use Internet to share health-related information by Medicare patients. Patients use the information they received from Internet to make decisions of their health care providers, choices of treatment and improve self management, the perceived accuracy, relevance of information are important. Hence, there is need for improvement of online information quality by means of providing filtered and peer reviewed content and the use of credibility rating systems [26].

Our results show that system security and privacy is positively associated with use of the Internet to share health information. Health-related information is private and sensitive personal information. Patients will only use the Internet for sharing health information if they are confident about the

safeguards in place to protect their medical records. Regulations like HIPAA do not apply to most websites and social networks. Policies must be created to involve patients in deciding who is involved in and how they are going to collect, use, and share their data.

Perceived health status is negatively related to Internet use for sharing health-related information. The more elderly patients under Medicare are more likely to share health-related information online if they feel they are in poor health condition or if it is difficult for them to take care of their health. Communication with other patients with similar conditions will likely help them find better ways to manage their heath.

VI. Conclusion

Online health-related information sharing among patients covered under Medicare is of high interest to the scholars given the prevalence, multiplicity and adverse nature of illness, and the variety of treatment options available. Use Internet for sharing health-related information is growing in popularity in online health communities. With the elderly population projected to double in the next 30 years, it will be important to study the factors contributing to their Internet use for health-related information. In this paper, by adapting the IS success model we have developed a conceptual model and investigated five factors that influencing the Internet use for health information — information quality, system security and privacy, physician patient communication, negative emotions, and perceive health status. The results show that education level does not significantly affect Internet use behavior.

VII. Limitations

There are several limitations of our study that will be addressed in future work. First, our dependent variable "Internet use" is a binary variable. Future studies should also look at the extent of use. Second, the exploration of the moderating effect of income level is also a possible future work.

APPENDIX A. MEASUREMENT ITEMS

	Items
Use Internet to Share Health Information	In the last 12 months, have you used the Internet to: Visited a social-networking site to read and share about medical topics? In the last 12 months, have you used the Internet to: Participant in a online support group for people a similar disease?
Information Quality 1	A5b. Based on your most recent search for information about health and medical topics, how much do you agree or disagree: You felt frustrated during your search for the information
Information Quality 2	A5c. Based on your most recent search for information about health and medical topics, how much do you agree or disagree: You were concerned about the quality of the information
System Security and Privacy 1	J3.How confident are you that safeguards are in place to protect your medical records from being seen by people who aren't permitted to see them?
System Security and Privacy 2	J4. How confident are you that you have some say in who is allowed to collect, use, and share your medical information?
Physician Patient Communication 1	In the past 12 months, how often did your health professional: Give you the chance to ask all the health-related questions you had?
Physician Patient Communication 2	In the past 12 months, how often did your health professional: Explain things in a way you could understand?
Physician Communication 3	In the past 12 months, how often did your health professional: Make sure you understood the things you needed to do to take care of your health?
Perceived Health Status 1	In general, would you say your health is?
Perceived Health Status 2	Overall, how confident are you about your ability to take good care of your health?
Negative Emotions 1	Over the past 2 weeks, how often have you experienced: Feeling down, depressed or hopeless?
Negative Emotions 2	Over the past 2 weeks, how often have you experienced: Feeling nervous, anxious or on edge?
Negative Emotions 3	Over the past 2 weeks, how often have you experienced: Not being able to stop or control worrying?

APPENDIX B. FACTOR LOADINGS AND ITEMS

	Negative Emotions (NE)	Perceived Health Status (PHS)	Information Quality (IQ)	Physician Patient Communication (PPC)	System Security and Privacy (SSP)
NE 1	0.9041	-0.3773	0.1677	-0.1128	-0.0526
NE 2	0.8989	-0.3949	0.1537	-0.1149	-0.033
NE 3	0.9000	-0.3496	0.1656	-0.1175	-0.0551
PHS 1	-0.349	0.8598	-0.1499	0.1654	0.0574
PHS 2	-0.3765	0.8859	-0.1758	0.2568	0.1445
IQ 1	0.1359	-0.1350	0.8322	-0.1127	-0.111
IQ 2	0.1769	-0.1882	0.9336	-0.1568	-0.0916
PPC1	-0.0949	0.1926	-0.1257	0.8448	0.1547
PPC 2	-0.1265	0.2406	-0.1526	0.9129	0.1566
PPC 3	-0.1189	0.2194	-0.1391	0.9187	0.1857
SSP 1	-0.0436	0.1017	-0.1018	0.1765	0.9147
SSP 2	-0.0503	0.1129	-0.0996	0.1608	0.8998

APPENDIX C. VALIDITY AND RELIABILITY

	PHS	PPC	IQ	SSP	NE	AVE	Composite Reliability	Cronbach's Alpha
PHS	0.8729					0.762	0.8649	0.6884
PPC	0.2442	0.8927				0.797	0.9216	0.8721
IQ	-0.1871	-0.156	0.8844			0.782	0.8774	0.7329
SSP	0.118	0.1861	-0.111	0.9073		0.823	0.903	0.7855
NE	-0.416	-0.1276	0.1799	-0.0516	0.9010	0.812	0.9283	0.8843

APPENDIX D. ASSESSMENT OF COMMON METHOD BIAS

To assess common method bias, we employed Harman's single-factor test measurement [33]. We performed an exploratory factor analysis to determine the number of factors that are necessary to account for the variance in the variables. If a substantial amount of common method variance is present, either a single factor will emerge from the factor analysis or one general factor will account for the majority of the covariance among the variables. The analysis produced five factors with Eigen values greater than 1.0. All of these factors collectively explained 79.51% of the variance of the data, with their values ranging from 8.75% to 30.18%. The first extracted factor accounted for 30.18% of the variance in the data, the second for 17.92%. This indicates that common method variance is not a serious concern.

The above test has been criticized for having insufficient sensitivity to detect a moderate or small level of the common bias effect [24]. We performed another test suggested by [33] and adopted by [21]. To assess method variance, each indicator is converted to a single-indicator construct, and all major constructs and the method factor become second-order constructs. A common method factor is included in the model, with this factor linking to all the single-indicator constructs [21]. For each single-indicator construct, the coefficients of its two incoming paths from its substantive construct and the method factor are equal

to the observed indicator's loadings on its substantive construct and the method factor. Common method bias can be assessed by comparing the variance of each observed indicator to its substantial construct and the method factor.

The results of this analysis for our study are presented in Table 8. The results show that the loadings of the method factor are insignificant in all of cases, and that the indicators' substantive variances are substantially greater than their method variances. Thus we can conclude that common method bias is not a serious concern [21].

TABLE II
COMMON METHOD BIAS ANALYSIS

Indicator	Substantive Factor Loading (R1)	R12	Method Factor Loading (R2)	R22
Information Quality 1	0.873*	0.7621	-0.037	0.0014
Information Quality 2	0.904*	0.8172	0.037	0.0014
System Security and Privacy 1	0.907**	0.8226	-0.001	0.0000
System Security and Privacy 2	0.908*	0.8245	0.001	0.0000
Physician Patient Communication 1	0.865*	0.7482	0.025	0.0006
Physician Patient Communication 2	0.901*	0.8118	0.019	0.0004
Physician Communication 3	0.912*	0.8317	0.004	0.0000
Perceived Health Status 1	0.925*	0.8556	-0.077	0.0060
Perceived Health Status 2	0.821*	0.6740	0.077	0.0060
Negative Emotions 1	0.884*	0.7814	0.011	0.0001
Negative Emotions 2	0.918*	0.8427	0.014	0.0002
Negative Emotions 3	0.902*	0.8136	-0.003	0.0000

* $p < 0.01$

APPENDIX E. USE INTERNET TO SHARE HEALTH-RELATED INFORMATION

	B	S.E.	Wald	Sig.	Exp(B)	95% C.I.for EXP(B)	
						Lower	Upper
Negative Emotion	.216	.161	1.795	.180	.806	.588	1.105
Physician Patient Communication	-.159	.192	.685	.408	.853	.586	1.242
System Security and Privacy	.694	.180	14.823	**.000**	.500	.351	.711
Information quality	.461	.170	7.390	**.007**	.630	.452	.879
Perceived Health Status	-.383	.167	5.255	**.022**	1.467	1.057	2.035
Constant	2.334	.184	160.970	.000	.097		

^ p < 0.10; * p < 0.05; ** p < 0.01

APPENDIX F. THE IMPACT OF EDUCATION ON USE INTERNET TO SHARE HEALTH-RELATED INFORMATION

	Lower Education Level B	S.E.	Wald	Sig.	Exp(B)	95% C.I.for EXP(B)		Higher Education Level B	S.E.	Wald	Sig.	Exp (B)	95% C.I.for EXP(B)		Z score for beta coefficient differences
						Lower	Upper						Lower	Upper	
Negative Emotion	.148	.362	.168	.682	1.160	.571	2.357	.376	.184	4.174	**.041**	.686	.478	.985	-0.56
Physician Patient Communication	-.130	.429	.092	.762	1.139	.491	2.639	-.258	.224	1.331	.249	.772	.498	1.198	0.26
System Security and Privacy	1.181	.502	5.523	**.019**	.307	.115	.822	.659	.201	10.746	**.001**	.517	.349	.767	0.96
Information quality	.116	.389	.089	.766	.891	.416	1.909	.562	.195	8.291	**.004**	.570	.389	.836	-1.02
Perceived Health Status	-.048	.366	.017	.896	1.049	.512	2.148	-.465	.191	5.929	**.015**	1.593	1.095	2.316	1.01
Constant	2.729	.558	23.904	.000	.065			2.269	.203	124.970	.000	.103			

^ p < 0.10; * p < 0.05; ** p < 0.01

REFERENCES

[1] Abbott, R. (2010). Delivering quality-evaluated healthcare information in the era of Web 2.0: design implications for Intute: Health and Life Sciences. Health Informatics Journal, 16(1), 5-14.

[2] Appari, A. and Johnson, E. (2010) Information security and privacy in healthcare: current state of research. International Journal of Internet and Enterprise Management, 6:4.

[3] Backhaus, K., Erichson, B., Plinke, W., and Weiber, R. (2003). Multivariate analysemethoden (10th ed.). Berlin: Springer, 2003.

[4] Bansal, G., Zahedi, F., & Gefen, D. (2010). The impact of personal dispositions on information sensitivity, privacy concern and trust in disclosing health information online. Decision Support Systems, 49(2), 138-150.

[5] Brodie, M., Flournoy, R., Altman, D. (2000). "Health information, the Internet, and the digital divide," Health Aff (Millwood).2000;19:255-265.

[6] DeLone, W.H., and McLean, E.R. 1992. "Information Systems Success: The Quest for the Dependent Variable," Information Systems Research (3:1), pp 60-95.

[7] Dickerson, S., Reinhart, A. M., Feeley, T. H., Bidani, R., Rich, E., Garg, V. K., & Hershey, C.O. (2004). "Patient Internet use for health information at three urban primary care clinics," Journal of the American Medical Informatics Association, 11(6), 499-504.

[8] Dutta-Bergman, M. 2003. "Trusted Online Sources of Health Information: Differences in Demographics, Health Beliefs, and Health-Information Orientation," Journal of Medical Internet Research (5:3), p. e21.

[9] Frost, J. H., & Massagli, M. P. (2008). Social uses of personal health information within PatientsLikeMe, an online patient community: what can happen when patients have access to one another's data. Journal of Medical Internet Research, 10(3).

[10] Frost, J. H., Massagli, M. P., Wicks, P., & Heywood, J. (2008). How the Social Web Supports patient experimentation with a new therapy: The demand for patient-controlled and patient-centered informatics. Paper presented at the AMIA Annual Symposuim Proceedings Archive.

[11] Giles, D. C., & Newbold, J. (2011). Self- and Other-Diagnosis in User-Led Mental Health Online Communities. Qualitative Health Research, 21(3), 419-428.

[12] Gill, P. S., & Whisnant, B. (2012). A qualitative assessment of an online support community for ovarian cancer patients. Patient Related Outcome Measures, 3, 51-58.

[13] Gutierrez, G. (2001). "Medicare, the Internet, and the future of telemedicine," Critical Care Medicine, 29, N144–N150.

[14] Hair, J. F., Anderson, R. E., Tatham, R. L., and Black, W. C. (1995). Multivariate data analysis with readings (4th ed.). Englewood Cliffs, NJ: Prentice Hall.

[15] Høybye, M. T., Johansen, C., & Tjørnhøj- Thomsen, T. (2004). Online interaction. Effects of storytelling in an Internet breast cancer support group.Psycho- Oncology, 14(3), 211-220

[16] Huang, M., Hansen, D., & Xie, B. (2012). Older adults' online health information seeking behavior. Paper presented at the Proceedings of the 2012 iConference, Toronto, Ontario, Canada.

[17] Jemal, A., Bray, F., Center, M. M., Ferlay, J., Ward, E. and Forman, D. (2011), "Global cancer statistics," CA: A Cancer Journal for Clinicians, 61: 69–90.

[18] Kaplan, G.A., and Camacho, T. 1983. "Perceived Health and Mortality: A Nine-Year Follow-up of the Human Population Laboratory Cohort," American Journal of Epidemiology (117:3), pp. 292-304.

[19] Kahai, S. S., and Cooper, R. B. (2003). Exploring the core concepts of media richness theory: The impact of cue multiplicity and feedback immediacy on decision quality. Journal of Management Information Systems 20(1): 263–299.

[20] Kisekka, V., Sharman, R., Singh, R., and Singh, G. "Misinformation in Healthcare Social Networks: Contributing Factors" (2011). AMCIS 2011 Proceedings - Paper 423.

[21] Liang, H., Saraf, N., Hu, Q., and Xue, Y. (2007). Assimilation of enterprise systems: the effect of institutional pressures and the mediating role of top management, MIS Quarterly 31(1): 59.

[22] Lee, S.Y., Hwang, H., Hawkins, R., and Pingree, S. (2008). "Interplay of Negative Emotion and Health Self-Efficacy on the Use of Health Information and Its Outcomes," Communication Research (35:3), pp. 358-381.

[23] Love, B., Crook, B., Thompson, C. M., Zaitchik, S., Knapp, J., LeFebvre, L.,. Rechis, R. (2012). Exploring Psychosocial Support Online: A Content Analysis of Messages in an Adolescent and Young Adult Cancer Community. Cyberpsychology, Behavior, and Social Networking, 15(10), 555-559.

[24] Malhotra, N. K., Kim, S. S., and Patil, A. (2006). Common method variance in IS research: A comparison of alternative approaches and a reanalysis of past research. Management Science 52(12): 1865–1883.

[25] Mauksch, L.B., Dugdale, D.C., Dodson, S., and Epstein, R. 2008. "Relationship, Communication, and Efficiency in the Medical Encounter," Archives of Internal Medicine (168:13), pp. 1387-1395.

[26] Metzger, M. J. (2007), Making sense of credibility on the Web: Models for evaluating online information and recommendations for future research. J. Am. Soc. Inf. Sci., 58: 2078–2091.

[27] Meingast, M., Roosta, R., and Sastry, S. (2006) Security and Privacy Issues with Health Care Information Technology. Proceedings of the 28th IEEE EMBS Annual International Conference. New York City, USA, Aug 30-Sept 3, 2006

[28] National Survey of Old Americans. 2005. "e-Health and the Elderly: How Seniors Use the Internet for Health Information," Keiser Family Foundation, Jan 2005

[29] Ong, L.M.L., de Haes, J.C.J.M., Hoos, A.M., and Lammes, F.B. 1995. "Doctor-Patient Communication: A Review of the Literature.," Social Science & Medicine (40:7), pp. 903-918

[30] Oh, H. J., & Lee, B. (2011). The Effect of Computer-Mediated Social Support in Online Communities on Patient Empowerment and Doctor–Patient Communication. Health Communication, 27(1), 30-41. doi: 10.1080/10410236.2011.567449.

[31] Petter, S. and Mclean, E., R. (2009). A meta-analytic assessment of the DeLone and McLean IS success model: An examination of IS success at the individual level. Information and management. (46): 159-166.

[32] Peng, C., J. and So, T., S., H. (2002). Logistic regression and reporting: A primer. Understanding Statistics 1 (1): 31–70.

[33] Podsakoff, P., MacKenzie, S., and Podsakoff, N. (2003). Common method biases in behavioral research: A critical review of the literature and recommended remedies. Journal of Applied Psychology 88(5): 879–903.

[34] Radin, P. (2006). "To me, it's my life": Medical communication, trust, and activism in cyberspace. Social Science & Medicine, 62(3), 591-601.

[35] Rice, R. E. (2006). Influences, usage, and outcomes of Internet health information searching: multivariate results from the Pew surveys. International journal of medical informatics, 75(1), 8-28.

[36] Ringle, C. M., Wende, S., and Will, A. 2005. SmartPLS 2.0. Hamburg, Germany: University of Hamburg

[37] Ruo, B., Rumsfeld, J., Hlatky, M., Liu, H., Browner, W., and Whooley, M. 2003. "Depressive Symptoms and Health-Related Quality of Life: The Heart and Soul Study," JAMA (290:2), p. 215.

[38] Stewart, M.A. 1995. "Effective Physician-Patient Communication and Health Outcomes: A Review," Canadian Medical Association Journal (152:9), pp. 1423–1433.

[39] Social Security Advisory Board. (2009). The Unsustainable Cost of Health Care. Retrieved from http://www.ssab.gov/documents/TheUnsustainableCostofHealthCare_508.pdf

[40] Tian, Y., & Robinson, J. D. (2008). Incidental health information use and media complementarity: A comparison of senior and non-senior cancer patients. Patient Education and Counselling, 71(3), 340–344.

[41] Valero-Aguilera, B., Bermúdez-Tamayo, C., García-Gutiérrez, J. F., Jiménez-Pernett, J., Vázquez-Alonso, F., Suárez-Charneco, A., ... & Cózar-Olmo, J. M. (2012). Factors related to use of the Internet as a source of health information by urological cancer patients. Supportive Care in Cancer, 1-8.

[42] Wang, R. and Strong, D. (1996) "Beyond Accuracy: What Data Quality Means to Data Consumers". Journal of Management Information Systems, 12(4), p. 5-34.

[43] Williams, J. (2010, May). Social networking applications in health care: threats to the privacy and security of health information. In Proceedings of the 2010 ICSE Workshop on Software Engineering in Health Care (pp. 39-49). ACM.

[44] Winzelberg, A. J., Classen, C., Alpers, G. W., Roberts, H., Koopman, C., Adams, R. E., ... & Taylor, C. B. (2003). Evaluation of an Internet support group for women with primary breast cancer. Cancer, 97(5), 1164-1173.

[45] Xiao, N., Sharman, R., Rao, H., Upadhyaya, S. 2011. "Factors Influencing Online Health Information Search: An Empirical Analysis of

a National Cancer- Related Survey," ISOM Workshop on Healthcare and IS, University of Florida

[46] Ziebland, S., Chapple, A., Dumelow, C., Evans, J., Prinjha, S., and Rozmovits, L. (2004). "How the Internet Affects Patients' Experience of Cancer: A Qualitative Study," BMJ (328:6), pp. 1-6.

Customized Behavioral Normalcy Profiles for Critical Infrastructure Protection

Andrey Dolgikh, Zachary Birnbaum and Victor Skormin

***Abstract*— Targeted cyber-attacks present significant threats to modern computing systems. Industrial Control Systems (ICS), Supervisory Control and Data Acquisition (SCADA) systems, and military networks are examples of high value targets with severe consequences of a successful attack. Anomaly based detection can address targeted attacks against critical infrastructure systems as attacks are likely to distort observable system activity. Typically, anomaly detection is performed on the level of system calls and is known to have shortcomings. Much better results have been obtained by performing behavioral analysis on the highest level of behavioral semantics, and are presented herein. Most critical computer systems serve a specific purpose and are expected to run strictly limited sets of software. Modeling their behavior by customized normalcy profiles facilitates a dependable anomaly based IDS. This technology could be further enhanced by the utilization of behavioral misuse detection performed in the similar fashion. An experimental IDS implementing the above approach is demonstrated.**

***Index Terms*—Intrusion Detection & Prevention, Behavior Based Intrusion Detection, Automatic Signature Generation, Critical Infrastructure Security.**

I. Introduction

TARGETED CYBER-ATTACKS can be deployed against high value objects within the national infrastructure to perform espionage and/or sabotage [1], [2]. The most popular malware detection schemes, dominated by the binary signature-based approach, are inherently incapable of addressing targeted, zero-day malware attacks as the attack is not represented by a binary sample in the database.

Behavioral analysis offers a more promising approach to malware detection because behavioral signatures are more obfuscation resilient than the binary ones. Indeed, changing behavior while preserving the desired (malicious) functions of a program is much harder than changing only the binary structure. Malware usually has to perform some system operations (e.g. registry manipulation) that are easy to observe, and these operations are difficult to obfuscate or hide. Therefore, malicious programs are more likely to expose themselves to behavioral detection. While a database of specific behavioral signatures is still utilized, its size and rate of increase are significantly lower than the binary signature database.

The behavioral detector has to be able to distinguish malicious operations from benign ones, which is often difficult and is often determined only by context or environment. The challenge of behavioral detection is in devising a good model of behavior, which has the necessary discriminative power and can be tuned to the target environment.

There are two behavioral detection mechanisms: misuse detection and anomaly detection. Misuse detection looks for specific behavioral patterns known to be malicious, while the anomaly based approach responds to unusual (unexpected) behavior. The advantage of anomaly-based detection is in its ability to protect against previously unseen threats; however, it usually suffers from a high false positive rate. Misuse detection is usually more reliable in terms of detection performance but has two major drawbacks. First, defining a set of malicious patterns (signatures) is a time consuming and error prone task that calls for periodic updating. Second, it cannot detect any malicious code that does not expose known malicious behavior patterns. Consequently, it seems logical to combine both detection mechanisms thus resulting in a highly dependable IDS technology.

The objective of this paper is the development of individualized defense mechanisms for particular computer systems that will successfully address the thread of targeted attacks. It presents a technology to be deployed within a reasonably small, limited access computer network running exclusively a set of approved legitimate applications. It is based on the concept of functionality that constitutes the highest level of behavioral semantics. This includes automatic extraction of a Customized Normalcy Profile (CNP) i.e. a library of functionalities fully describing normal operation (behavior) of the network. The CNP is to be utilized as the major component of an anomaly based behavioral IDS. Anomalies, indicative of an attack, will be detected as occurrence of functionalities not present in the CNP, as an anomalous realization of a legitimate functionality, or as an anomalous timing/ frequency of execution of a functionality. The system can be upgraded by the inclusion of a misuse-based detection module utilizing a library of known malicious functionalities (stealing user credentials, backdoor, deletion of system files, prohibited operator actions, etc.).

The described technology is intended for deployment for the detection of targeted cyber-attacks against high value targets such as power plants, government installations, banks, etc.

This material is based upon work supported by the Air Force Office of Scientific Research (AFOSR) under Award No. FA9550-12-1-0077.

A. Dolgikh is with Binghamton University, Binghamton, NY 13902 USA (e-mail: adolgik1@binghamton.edu).

Zachary Birnbaum is with Binghamton University, Binghamton, NY 13902 USA (e-mail: zbirnba1@binghamton.edu).

Victor Skorminis with Binghamton University, Binghamton, NY 13902 USA (e-mail: vskormin@binghamton.edu).

II. Nature of Cyber Attacks Against ICS

Unlike physical attacks, cyber-attacks are perpetrated through a computer governing the operation of a physical system. The attacks could be classified as follows.

- Conventional self-propagating computer malware (worms, viruses and trojans). Such attacks typically disable the process control computer (along with many other hosts) by causing wide spread computer epidemics. In a SCADA environment such attacks are caused intentionally or unintentionally by personnel's non-compliance with the existing regulations. Such attacks can be addressed by the commercial antivirus tools.
- Targeted attacks as exemplified by the StuxNet worm. This type of attacking malware is uniquely tuned to the operational environment of a particular computer or local network. The malware is designed to peacefully coexist with the host performing benign exploratory functions ("low and slow attacks") until their last, malicious functionality is implemented. In the SCADA environment, malicious functionality may include altering parameters of the control law causing system instability, scaling magnitudes of the feedback or reference (set point) signals thus resulting in unacceptable operational regimes of the physical system. Technically, these attacks could be launched from the Internet, however in known instances they were launched from a USB drive. By their very nature, these attacks are "zero day attacks" that cannot be detected by a conventional antivirus.
- Attacks against industrial control equipment such as Programmable Logic Controllers (PLCs), Human-Machine Interfaces (HMIs), smart devices for distributed control, etc. Industrial control networks consist of multitude of interconnected devices running various specialized software. By subverting such devices, the attacker can achieve unprecedented level of control over industrial system. As it was shown by Stuxnet, it is possible to subject controlled equipment to dangerous and destructive operational regimes while creating a full illusion of its normal operation for personnel [2].
- Attacks against sensor networks. Many large-scale industrial processes are spatially distributed and are monitored by a network of sensors. In many instances, industrial processes require monitoring of the environmental factors that also involves a sensor network. Individual sensor data is fed into a computer and is subjected to individual or integrated processing. Sensor networks could be attacked by specially designed electric and sonic signals, electro-magnetic and thermal fields, etc.
- Physical damage of the equipment does not qualify as a cyber-attack. However, as far as the equipment in question is a part of the control loop along with digital controllers, physical damage would cause anomalous operation of the computer system that could cause further damage comparable with effects of a cyber-attack.

III. Industrial Control Systems

ICS networks usually have complex topology and consist of many components interconnected with proprietary protocols and include SCADA systems, industrial communication protocols and controllers. SCADA (supervisory control and data acquisition) generally refers to computer systems that monitor and control industrial processes [3].

The major components of a SCADA system are:

- A human interface which allows a human operator to observe and control the ongoing process.
- A supervisory system that collects data from the subsystems and sends commands back to them.
- Remote terminal units (RTUs) which connect to sensors and actuators in the process. RTUs send sensor data to supervisory system and convey commands from supervisory system to actuators.
- Controllers or programmable logic controllers (PLCs) perform local control actions according to the algorithm specified by a programmer. Programmable controllers range from very small to quite powerful computing units and can perform a large number of functions. Controllers are directly interfaced with sensors and actuators via industrial network protocols.
- Communication infrastructure connecting the supervisory system to the RTUs and PLCs.

A very important component not included in the above list is the SCADA support and on-going development environment. The SCADA development environment provides tools to keep up with a changing environment, requirements or structure of the system, and upgrades. SCADA development tools allow a change or reconfiguration in the behavior of any components listed above. This is usually done through special interfaces available to developers.

As shown in Fig. 1 there are two common ways to reach the ICS system, through enterprise networks, and through development interfaces via SCADA developer's computers. Both ways have advantages and disadvantages from the attacker's point of view. Enterprise vector of attack provides attacker with known environment and stable access to the system. On the other hand enterprise systems are usually well monitored and logged that makes a successful mounting of the attack difficult.

For this reason the most dangerous and resilient known attacks on SCADA systems were deployed through development/maintenance interfaces [2], [4], [5]. Unfortunately, most SCADA software and control hardware is not build with security in mind. Security and logging functionality are very often overlooked by the system developers. While a successful development attack vector requires familiarity with the specifics of the targeted system, it is not a problem for a determined attacker; today, required specifics are available from the Internet [6]. Another

advantage of attacks through developer/maintenance interfaces is that they do not leave many traces behind. A modified controller software may work for years waiting for malicious commands.

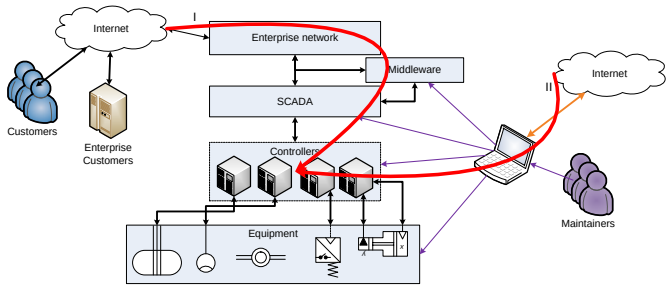


Fig. 1. Attack vectors of an industrial complex.

Industrial control systems expose several properties that complicate the development of effective protection mechanisms [4]:

- Many system components are highly sensitive to the formatting of data and network packets and show intolerance to timing issues. Building security software meeting such stringent criteria is a very complicated task.
- The hardware/software update cycles are extremely long. Typical field device lifetime ranges from 5 to 20 years. This means that average field device is at least ten years outdated. Ten years is a very long time in the computing and security industry. Computing power imbalance between field devices and potential attacker might be tremendous. Long update cycles also mean slow reaction to vulnerability disclosure. Very often ICSs run software which is no longer supported resulting in fundamental inability to patch newly discovered vulnerabilities.
- Redundancy and no downtime (even momentarily) requirements pose additional problems for update cycles. Updating software or hardware must go seamlessly without interruption of major functionality of the system.
- Huge variety of interconnection protocols and custom implementations in industrial control field. This severely complicates the development of security solutions and decreases the probability of early problem exposure.

At the same time industrial systems have several properties that are likely to be exploited for building dependable defense mechanisms:

1. Operation profile for the most industrial systems is very static in nature. The set of functionalities seen during normal system operation do not change with respect to time The white listing approach can use this property.
2. Industrial control systems evolve very slowly in comparison to the rest of computing industry. This means that implemented solutions can work for years allowing more time for the development of customized solutions.
3. Industrial systems usually evolve and do not exhibit sudden changes in architecture or operation profile. This can be used for identification and authentication of introduced changes. If the change comes from unknown source or looks different from expected it can always be aborted.

The properties of industrial control systems listed above justify the deployment of the security monitoring/protection approach suggested herein.

IV. Approach

The problem of detecting unknown behavior can be split in two phases:

During the off-line phase, we observe a stream of system level events for a time period sufficient to recover the majority of application cases. This accumulated data is used to build a model of the known behavior of the system. The compilation of the behavioral model can be done off-line with extensive use of computing resources.

During the on-line phase, we match an observed stream of events with models of known behavior. If the stream deviates from the behavior predicted by the model we declare an anomaly. In order to be practical, this step should be performed with low overhead.

In this paper we will focus on the process of assembling a behavioral model from the continuous stream of the system calls. This intention is not new, many attempts were made to create models of normal and malicious behavior. Different types of mechanisms: finite automata, context free languages, Markov models, etc. were utilized to monitor data dependencies and system call dependencies on order to grasp the complex relationship between system calls and data they operate on. In our view, the limited success of these efforts is attributed to the fact that they typically lead to elusive and unstable models which only remotely reflect the behavior of the program. In this paper we utilize a behavior modeling approach operating on the highest level of behavioral semantics, the level where behavior could be directly associated with the specific goals of the software developer.

V. Behavioral Representation

The behavior refers to the actions performed by the program with respect to its environment. The environment of any program in a computer system is controlled and managed by operating system kernel. Windows OS organizes the environment using OS objects: file, memory section, thread, mutex, etc. For a program to sense or modify the environment a request to kernel must be issued. The request to OS is issued by invoking a system call with desired parameters.

For example, system call ZwOpenKey has the following parameters: KeyHandle, DesiredAccess, ObjectAttributes.

KeyHandle is a handle to the opened key. Generally, a handle is a context specific reference/tag to the OS managed object. It allows referencing the same object in a sequence of system calls. DeasiredAccess is the value that determines the requested access to the object. ObjectAttributes is a structure that specifies the object name and other attributes. Unlike handles, the names are system wide object identifiers.

Handles and object names are especially important for observing the behavior since handles and names directly correspond to some OS objects. Additionally, equal handles or names provide clear indication that system calls were issued for the same OS object.

Monitoring system calls along with parameters provides system-wide view on behavior. In order to reason about the behavior and especially about anomalous changes in the behavior we suggest such a model that allows us to capture the normal structure of operations over OS objects.

Formally, the model is a vertex-edge labeled graph which is constructed from the stream of system calls,

$$Gm = (V, E, Fv, Fe) \quad (1)$$

Where
V – set of vertices,
E – set of edges,
F_v - mapping from V to set of system calls S.
F_e - mapping from E to set of data links D.

It is worth noting that the set of system calls S is small and well known. On the other hand elements of D represent all possible values of parameters of system calls. Therefore the set of data links D is unknown beforehand and very extensive. Fortunately, this does not pose a difficult problem since majority of the important parameters take values from the small subset of D.

The graph G_m can be built from the stream of system calls according to the following rules:

1. Labeled vertex v_s is added to $\mathbf{G_m}$ for each issued system call *s*.
2. Labeled edge e_d from v_i to v_j is added when v_i and v_j share the same data *d* and one of the following:
   (a) v_i has *d* as the output and v_j takes *d* as the input
   (b) v_i was registered before v_j

For example, calls S1, S2 in (Fig. 2. a) have a common parameter C. In the resulting graph (Fig. 2. b) nodes corresponding to calls S1, S2 are connected with the directed edge C. Nodes S2, S3 are connected with an edge labeled C.

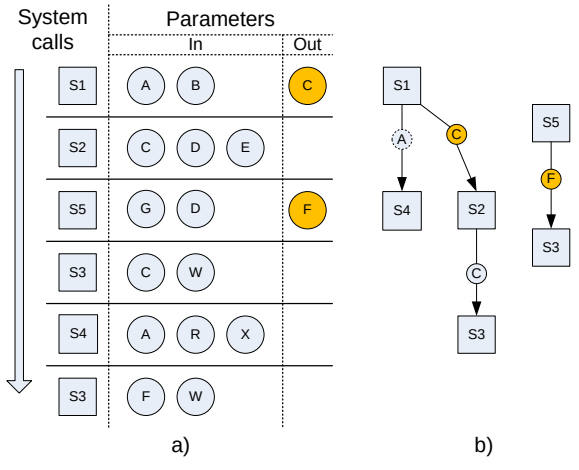


Fig. 2. Links in the graph.

The described process transforms the stream of system calls into a stream of graphs. Due to the high volume of system calls these graphs usually grow to unmanageable sizes, primarily due to repetitive/cyclic actions. On the other hand, repetitive occurrences of a single system call or some graph substructure do not provide additional dependency information. Therefore it is beneficial to detect and eliminate repetitive structures (Fig. 3).

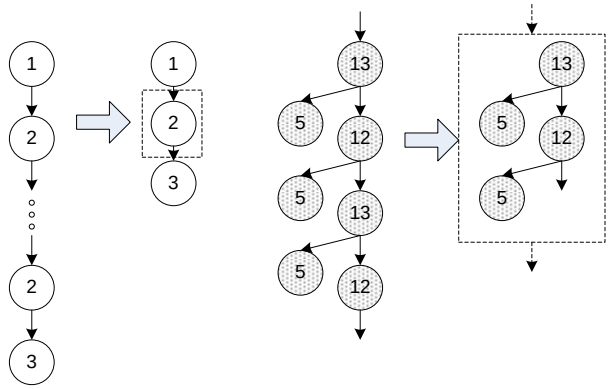


Fig. 3. Graph compression.

There are several graph compression algorithms suggested in [7], [8]. These algorithms were developed for problems quite different from system call graph compression. The frequent substructures search algorithm from [7] does not scale up to graphs with tens of thousands of nodes. The algorithm described in [8] is more efficient but still too "expensive" for large number of nodes. In addition, the semantics of graph compression described in these papers cannot be directly applied to system call graphs.

The graphs featured in Fig. 4 represent overwhelming majority of the system calls graph types observed. As one may see, they have a very simple structure reflecting repetitive operations over one or two OS objects. In spite of simplicity they result in a very heavy performance penalty for general graph compression algorithms.

Most of huge system call graphs are generated by simple cycles, therefore such traces need to be effectively recognized and eliminated. For system call graph matching we generally do not care how many times some substructure is repeated. It gives us an opportunity to relax compression accuracy and have irrelevant data disappear. In particular, we replace linear repetitive parts of the graph with one representative component.

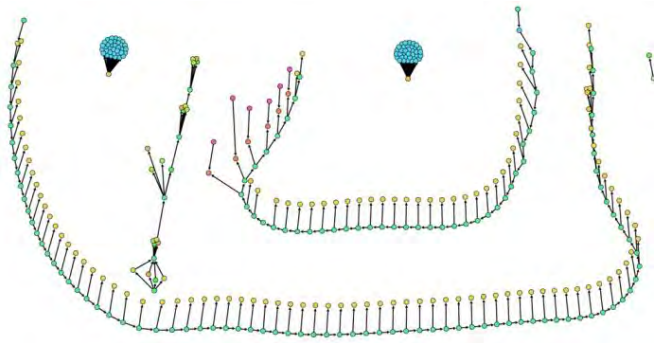


Fig. 4. Typical system call graphs.

Lossy graph compression adapted to system call graphs allows for much faster algorithm and better compression of the graph.

VI. ALGORITHM

Our graph compression algorithm is based on the Graphitour algorithm [8]. This algorithm is proved to be instrumental for biological and genetic data processing [9]. It is not the fastest algorithm known to date [10] but it has certain properties that we exploit to our benefit. It is described in detail in [11].

The procedure was evaluated on execution traces obtained from several benign and malicious programs running on Windows XP. System call traces were recorded from our driver which intercepted system calls with their arguments by hooking into the SSDT table. To evaluate our approach in general, without any prior knowledge of the importance of individual system calls for security, we intercepted all of the calls referenced by the service table, except a few for which we could not find the correct specification of input arguments. We used the specially developed driver to obtain execution traces from several malicious and benign applications.

Malicious programs were obtained from the Offensive Computing website [12] and included malware samples of different types and from several families. Benign programs were selected to represent a typical user setup. We joined the obtained samples into three testing traces so that each trace consisted of several malware types and several benign programs.

Since we monitored all of the system calls, the size of execution traces grew rapidly with time, quickly exceeding 10GB for large traces. Therefore we used traces obtained only for a limited amount of time, ranging from 1 minute to 20 minutes in the case of longest execution trace. The compression was applied to the entire graph that could have over hundred thousand nodes. In the future, some incremental, real time compression schemes could be developed allowing the processing of much longer traces. The results could be seen in Table 1.

TABLE I
TESTING/VALIDATION OF THE FUNCTIONALITY EXTRACTION PROCEDURE

Trace #	Number of system calls	Number of unique graph components	Number of detected functionalities	Number of malicious functionalities detected
1	6927937	1047	341	23
2	3704217	862	307	21
3	20719	217	49	9

Two functionalities extracted from the real system call data by the application of the described procedure can be seen in Fig. 5. Component #169 on the left represents typical interaction with Windows registry. Component #171 corresponds to remote thread injection functionality. Such functionality is rarely used by legitimate software. And it is no surprise that it was obtained from the data containing malware execution traces.

VII. NORMALCY REPRESENTATION

Rules mined by the Graphitour algorithm over a substantially large dataset of system calls representing execution of legitimate software result in a set of functionalities that can be perceived as behavioral signatures of normal system operation. It is important that the described procedure runs without the involvement of human operator, i.e. the functionalities are extracted automatically.

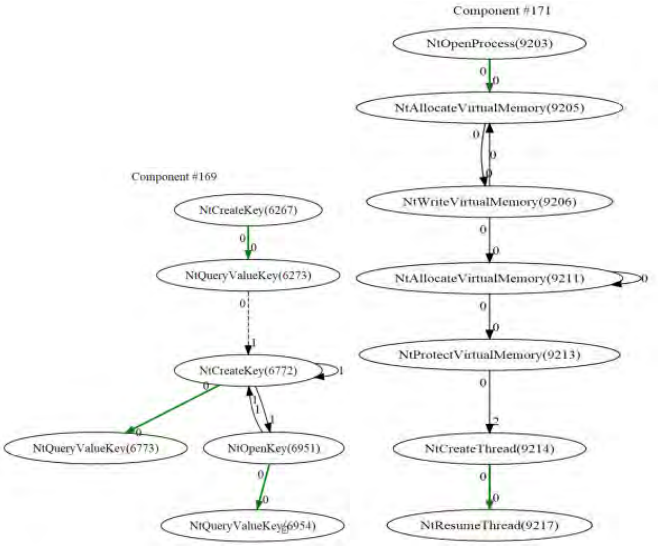


Fig.5. Functionalities extracted from "real" system call data.

The availability of behavior models, i.e. legitimate or malicious functionalities marks the completion of the off-line phase of the modeling effort. The on-line phase calls for the most efficient technology for the functionality detection. We have shown previously that Colored Petri nets (CPN) present a very efficient tool for performing this task [13]. Translation from Graphitour rules (essentially graphs) into CPNs is relatively straightforward. Each node of the rule-graph corresponds to a place in CPN. Incoming edges of the nodes are fused into transitions. Guard expressions of transitions are obtained from edge link types (see Fig. 6).

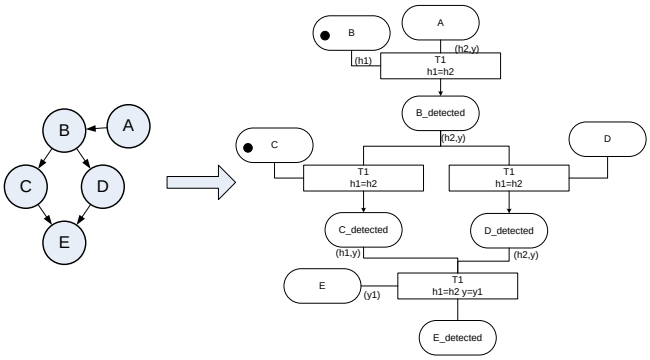


Fig. 6. Translation of Graphitour rules into CPN.

A set of CPNs obtained from Graphitour mining are results that cover all possible legitimate activities for a local network facilitates the anomaly detection. In a stable setting this makes our detector much more reliable and efficient than the SUMMARIZE-MINE mechanism described in [14].

The customized normalcy profile is perceived as a set of automatically extracted functionalities, accompanied by the frequencies of their execution. Unlike a public network providing services to a wide community of users, networks of

a "high value facility" are expected to demonstrate a very rigid set of functionalities and their frequencies. The detection of unseen earlier, not necessarily malicious activity, and/or mere changes in the execution frequencies of the functionalities would indicate an attack. Fig. 7 illustrates the described IDS concept.

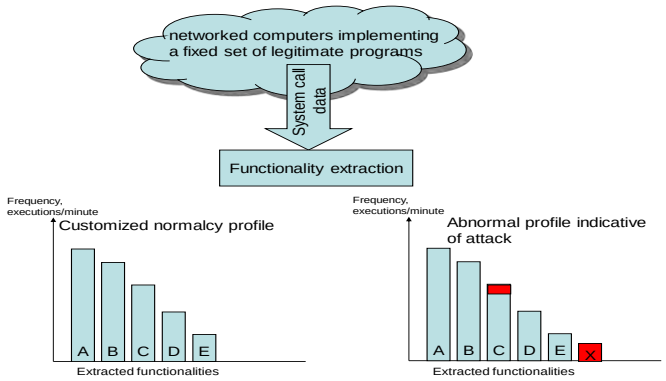


Fig. 7. IDS utilizing a customized normalcy profile.

The implementation of the described approach includes the development and periodic updating of the normalcy profile, and the on-going task of functionality extraction, detection of known malicious functionalities, and the anomaly detection in network operation.

VIII. Resultant IDS

The objective of our research is to provide individualized defense mechanisms for particular computer systems that will successfully address the threat of targeted attacks. The resultant IDS shall be capable of detecting "low and slow" targeted attacks at the earliest stages of their deployment as "extracurricular" activities of the system processes. The detection decisions are made on the highest semantic level of process behavior, where the division between malicious and benign behavior is well defined, thus the number of false positives and false negatives are drastically minimized.

- The proposed approach does not utilize binary signatures of malware. Instead it is based on behavioral signatures that are more obfuscation resilient than the binary ones.
- Unlike general purpose anti-viruses, our approach results in a customized defense mechanism that fully addresses the uniqueness of the specific high value object. The customization enables the detection of "low and slow" targeted attacks that are expertly tuned to the network environment.
- The detection decisions are made on the level of functionalities where malicious and benign behavior patterns are well separated.
- The IDS combines anomaly based detection with misuse detection by utilizing two sets of behavioral signatures.
- While all exploits leave a noticeable trace at the level of system calls, the task of writing an undetectable exploit becomes very complicated. Privilege escalation in that case might be rendered unachievable because it relies on modification of program environment which cannot go unnoticed through graph matching algorithm.

Consequently, the proposed effort will result in a highly dependable resident security system performing real time monitoring of the network behavior, capable of detecting anomalies indicative of information attacks.

IX. Experimental Implementation

For the purpose of demonstration we deployed the developed technology to detect attacks against a vulnerable system.

The host under attack is represented by the Metasploitable Virtual Machine [15]. The Metasploitable is an operating system package which comes preconfigured with many vulnerable servers and services. It is an ideal playground for testing intrusion detection systems.

We use the Metasploit framework to launch the attack [16]. Metasploit comes preconfigured with tools for network identification, vulnerability scanning and penetration testing. It provides means to find exploitable vulnerabilities and mount attacks against them. We used Backtrack Linux to host Metasploit in our demonstration [17].

Since our IDS works with graphs we use Gephi graph visualization software [18]. It allows us to show in real time system call data and visualize previously captured and saved trace files. The use of this facility enables us to observe both benign processes and malicious exploits developing in real time. It should be noted that the particular configuration and location of the individual graph components is the function of the visualization software and is irrelevant.

The Normalcy Profile of the Metasploitable VM is defined under no load. Running the test under no load clearly confirms our assumption that a stable normalcy profile of the no-load system in question exists and could be detected. Generating a trace with over 100,000 graphable components we established our base data set. The data shows that the number of newly detected graph components tends to stabilize with time thus confirming our assumption. Indeed, twenty components were successfully indentified and for the remainder of the test no additional components were detected.

Unlike the baseline component library which molded system behavior under no load we needed additional profiles under load. The Metasploitable VM comes with many exploitable servers, including an easily exploited SAMBA server. SAMBA is responsible for file and print sharing services and is found in most industrial or enterprise environments.

During the demonstration we expose SAMBA server process to typical load such that copy, move, delete files and folders. Once the analysis of the SAMBA trace was complete a normalcy profile reflecting normal samba operation contained 24 components was established (see Fig. 8). Adding more new functionalities to the profile by monitoring system operation was impossible.

Once it was confirmed that the stable normalcy profile of a network is a reality, we begin testing the IDS using real world malicious exploits.

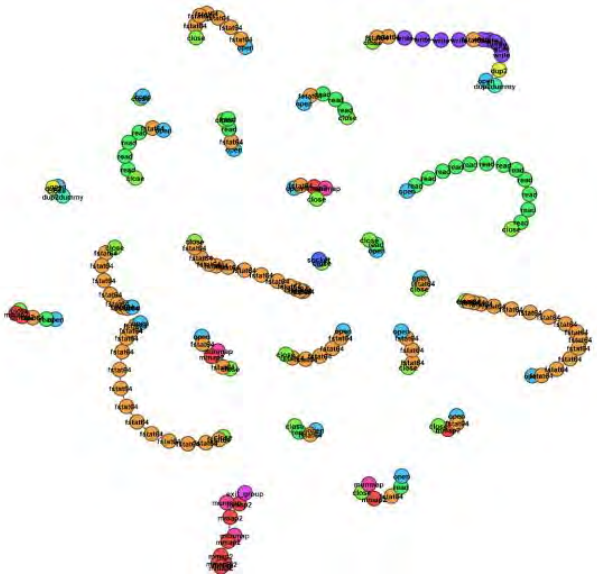


Fig. 8. Component Database for Samba Server.

The Metasploit Framework includes an exploit for SAMBA versions 3.0.20-3.0.25rc3 allowing an attacker to execute arbitrary commands. Fig. 9 features a graph trace for Samba under the Metasploit attack. Enlarged red nodes correspond to components not found in our normalcy profile. These components manifest additional anomalous functionality not observed before on the Samba server.

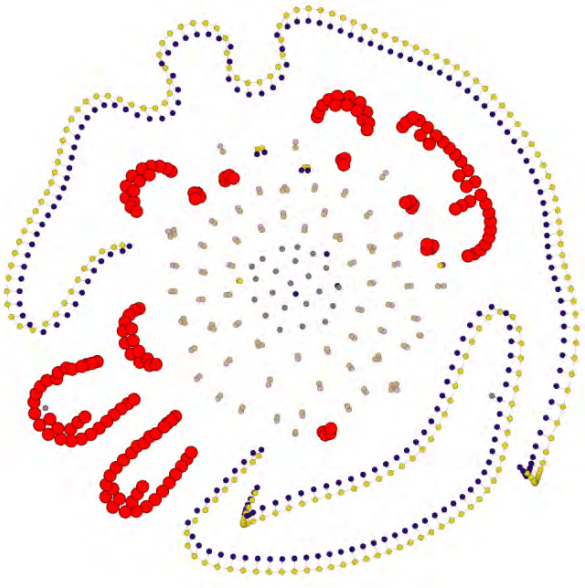


Fig. 9. Successfully detected anomalies (enlarged red nodes).

The anomalous components found during the Samba attack are invaluable for the identification of such an attack. Fig. 10 features ten anomalous components extracted in the experiment described above.

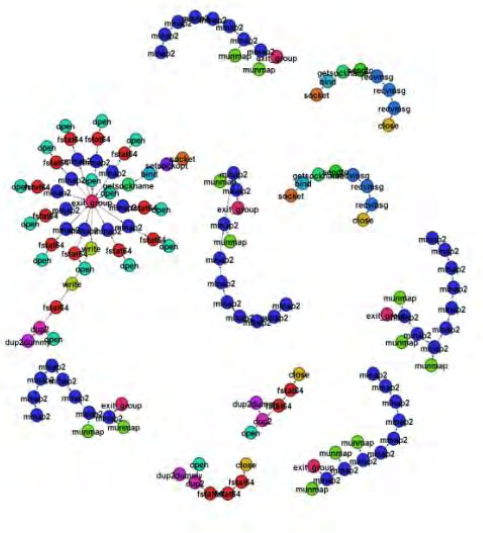


Fig. 10. Identified Malicious Components.

Our demo consists of videos taken off the computer terminal performing real time operations. They show individual system calls being detected and connected into graph components based on their attributes.

The first part of the video features real time normalcy profile generation followed by real time comparison of the profile against normal operation of the machine. At this point it is clear that we do not detect anomalies and the profile is working as desired.

The second segment of the video includes an attack scenario that is successfully detected by the IDS. Sample video can be found at [19], [20], [21].

An equally promising is the application of the described approach for 'health monitoring" of individual processes/applications that could subjected to targeted attacks. The selection of the process in question is achieved by the utilization of the relevant attributes of system calls.

X. ACKNOWLEDGEMENT

The authors are grateful to Dr. Robert Herklotz for supporting this effort.

REFERENCES

[1] Percoco, N., Ilyas, J.: Malware Freakshow 2010: White paper for Black Hat USA, (2010)
[2] Falliere, N., Murchu, L., Chien, E.: W32.Stuxnet Dossier: Symantec security response version 1.4, (2011)
[3] Online, SCADA: wikipedia.org/wiki/SCADA, accessed on December 30, 2011
[4] Keith Stouffer, Joe Falco, Karen Scarfone, Guide to Industrial Control Systems (ICS) Security, NIST special publication 800-02, June 2011
[5] Marshall Abrams, Joe Weiss, Malicious Control System Cyber Security Attack Case Study, NIST, 2008
[6] Jonathan Pollet "Electricity for Free? The Dirty Underbelly of SCADA and Smart Meters" Black Hat USA, 2010
[7] Inokuchi, A., Washio, T., Motoda, H.: An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data. In Proceedings of the 4th European Conference on Principles and Practice of Data Mining and Knowledge Discovery (PKDD), pp.13–23, (2000)
[8] Peshkin, L.: Structure induction by lossless graph compression. In Data Compression Conference, DCC, pp.53-62, (2007)

[9] Hayashida, M., Akutsu, T.: Comparing Biological Networks via Graph Compression. In Symposium on Optimization and Systems Biology, 2009
[10] Choi, Y., Szpankowski, W.: Compression of Graphical Structures: Fundamental Limits, Algorithms, and Experiments. In IEEE Transactions on Information Theory, 2012
[11] A. Dolgikh, T. Nykodym, V. Skormin, and Z. Birnbaum, "Using Behavioral Modeling And Customized Normalcy Profiles As Protection Against Targeted Cyber-Attacks", Proceedings MMM-ACNS October 17, 2012, St. Petersburg, Russia.
[12] Offensive Computing, http://offensivecomputing.net/ accessed in Nov 2011
[13] Dolgikh, A., Nykodym, T., Skormin, V., Antonakos, J.: Colored Petri nets as the enabling technology in intrusion detection systems. In Military Communications Conference, MILCOM 2011, pp.1297-1301, (2011)
[14] Chen, C., Lin, C.X., Fredrikson, M., Christodorescu, M., Yan, X., Han. J.: Mining graph patterns efficiently via randomized summaries. In Proceedings VLDB Endow, Vol.2, no.1, pp.742-753, (2009)
[15] Online, Metasploitable, http://sourceforge.net/projects/metasploitable/, Accessed Dec 2012.
[16] Online, Metasploit, http://www.metasploit.com/, Accessed Dec 2012.
[17] Online, Backtrack-Linux, http://www.backtrack-linux.org/. [Accessed 6 December 2012].
[18] Online, Gephi, Available: https://gephi.org, Accessed Dec 2012.
[19] Online, https://docs.google.com/open?id=0B5QdGhtUiP_AUjJMeWFET2draTQ
[20] Online, https://docs.google.com/open?id=0B5QdGhtUiP_ANXhXajQwUnY4cDA
[21] Online, https://docs.google.com/open?id=0B5QdGhtUiP_ANmdFZmxXc25IV0U

Consumer Adoption of Smart Metering Technology

Merrill Warkentin, Sanjay Goel and Philip Menard

***Abstract*— In response to escalating energy consumption through the world, utility companies have begun implementing smart meter technology (SMT) as a way to decrease overall consumption and reduce operating costs, such as maintenance of secondary power stations only functioning at high-demand times. Benefits are apparent for both consumers and utility companies, but consumers have concerns related to the privacy of their electrical usage data and giving the utility company control over their appliances. The present research uses the factorial survey method to examine customers' intentions to adopt SMT given specific circumstances which may mitigate the amount of privacy or control customers have over their electrical usage data and appliances. The findings suggest that although avoiding brownouts is a significant benefit to energy consumers, concerns over control and information privacy are significant as well. The results of this research should provide utility companies with an understanding of their customers' apprehensions and offer them insight as to how this new technology should be presented to customers in a way that will maximize adoption rates and minimize concerns. Based on consumers' perceptions of how electrical usage data may be shared, utility companies should also be informed on how to build fair data usage policies that openly address customer concerns related to adopting SMT.**

***Index Terms*—Smart metering technology, SMT, Smart Grid, Information privacy concerns, Technology adoption, Psychological ownership**

I. Introduction

ENERGY CONSUMPTION has been on an upward trajectory for the last several years. Consumers are concerned with minimizing this consumption as a means to reduce costs, as well as achieve energy independence and contribute to environmental and conservation initiatives [11]. The smart electric grid has emerged as a viable way to reach these goals by utilizing the existing infrastructure of the power grid as a communications network for utility companies. This emerging technology, which will be gradually implemented by utility companies in the near future, is intended to incrementally increase the efficiency of distribution and production [9].

As part of this initiative, smart metering technology (SMT) will be incorporated locally for each consumer in two installation varieties: an external meter with increased functionality or as a smart system which allows communication with individual appliances [1]. The degree of monitoring available through SMT should eventually result in a reduction in operating costs for power companies and better service for consumers, as well as contributing to environmental endeavors and the "green" movement [11].

One of the caveats consumers must consider, however, is electricity utilities' ability to actively power down non-critical appliances in certain households in order to negate the need for using secondary peak load demand plants. This capability will likely encounter some opposition by consumers, especially if there are no apparent monetary benefits [6].

While loss of control is an important consideration, consumers may also be concerned with the potential misuse of the electrical usage data being captured, including metadata produced by SMT [2], [4], [6], [7]. Several other consumer concerns have been identified, including illegal usage (burglars determining when houses are unoccupied), commercial usage (profiling by marketers), and law enforcement usage (detection of illegal activities) [8]. Utility companies could also become possible targets for hackers pursuing data regarding a large number of consumers [3].

While the association of information privacy concerns and technology adoption has been previously studied, this relationship has not been explored within the context of SMT acceptance. This study is focused on the following research questions: (1) how do consumers' concerns about information privacy influence their intentions to adopt smart metering technology?, (2) do perceived benefits of adopting smart metering technology alleviate consumers' concerns associated with information privacy?, and (3) does psychological ownership of electrical usage data affect consumers' information privacy concerns?

II. Method

To study the influence of ownership, information privacy concerns, trust, risk, social influence, and perceived benefits on a consumer's intention to adopt SMT, the experimental factorial design via utilization of scenarios was chosen as the appropriate method for this study. The factorial survey differs from typical scenario-based surveys in that textual elements within the scenario are experimentally varied [5], [10].

This paper was submitted on March 5, 2013.

Merrill Warkentin is Professor of MIS and the Richard Puckett Notable Scholar at Mississippi State University. Mississippi State, MS 39762 (e-mail: m.warkentin@msstate.edu).

Sanjay Goel is an Associate Professor in the Information Technology Management Department (School of Business) at the University at Albany, SUNY, Albany, NY 12222. He is also the Director of Research at the New York State Center for Information Forensics and Assurance at the University. (e-mail: goel@albany.edu).

Philip Menard is a PhD student in Information Systems at Mississippi State University. (e-mail: prm121@msstate.edu).

The following latent constructs were measured in our instrument with multi-item scales: psychological ownership, information privacy concerns, trusting beliefs, risk beliefs, social influence, attitude toward smart metering technology, and behavioral intent. Each item is measured using a five-point Likert scale, and all items were fully anchored from "strongly disagree" to "strongly agree."

Our scenarios were manipulated by embedding independent variables into the scenarios as manipulated values for perceived benefits, meter invasiveness, and data sharing. Each respondent was exposed to three different versions of the scenario. Following each scenario, respondents were presented with items measuring intention to adopt smart metering technology. After participants were exposed to the scenarios, the respondents were assessed on perceptions of psychological ownership, information privacy concerns, trusting beliefs, risk beliefs, and social influence, as well as general demographics questions, including age, gender, computer experience, education level, prior experience with personal privacy invasions, and exposure to news related to information privacy violations.

III. Discussion

Adapted from UTAUT, the impact of social influence indicates that consumers care about the perceptions and opinions of influential people in their lives with regards to SMT adoption. This particular facet of the UTAUT model is applicable in the SMT context. The specific implementation of smart meters and their associated data sharing policies also had a significant influence on behavioral intent in this study. Meter invasiveness had a significant negative influence on behavioral intent, showing that as the consumer cedes more control and information to the utility company, he or she is less likely to adopt SMT. Sharing electrical usage information with the utility company, the government, and marketers each had a significant negative effect on behavioral intent. The strength of the relationships between government access and marketer access with behavioral intent was especially strong, indicating that while consumers are apprehensive about sharing usage information with their utility companies, they may be even more cautious doing so when there is the possibility of that information being shared with the government or marketers.

Trusting beliefs and risk beliefs had a significant effect on behavioral intent. The addition of psychological ownership was strongly significant. Psychological ownership explained a fairly large amount of variance for both information privacy concerns and risk beliefs.

Respondents were also assessed on their individual perceptions of specific benefits associated with SMT, as well as general concerns toward hackers obtaining personal information. The only benefit that had a significant impact on behavioral intent was avoiding brownouts. This finding is interesting, as meter invasiveness was negatively significant. Our results show that while consumers are concerned about losing a degree of control over their appliances, they are interested in avoiding brownouts, which are mitigated through the utility company's selective power allocation via smart meters. When communicating the benefits of SMT, utility companies may need to emphasize brownout avoidance as a key benefit in order to convince consumers that SMT is ultimately favorable.

The findings provided by this study should enhance utility companies' understanding of their customers' concerns and equip them with knowledge on how to present this emerging technology to their customers in a manner that will maximize adoption and alleviate concerns. Based on consumers perceptions of sharing electrical usage data, utility companies may also be informed on how to construct fair data usage policies that directly address consumer apprehensions associated with adopting SMT.

IV. Conclusion

Information privacy will continue to be a widely discussed area of research, especially as smart devices become more ubiquitous in consumers' lives. However, some users may have concerns about the type of data that is being collected by various smart devices. Determining consumers' greatest concerns about collecting electrical usage information and creating measures to help protect this data will help foster greater adoption of SMT. Implementing fair usage policies and communicating these procedures to energy consumers should alleviate customers' fears and empower them to learn about the potential risks, as well as the benefits, of adopting smart meters. Emphasizing the avoidance of brown-outs or rolling blackouts may also be helpful in convincing energy consumers that SMT is truly beneficial and worthwhile.

References

[1] Darby, S., "Smart metering: what potential for householder engagement?" *Building Research & Information*, vol. 38, no. 5, pp. 442–457, 2010.

[2] Goel, S., Bush, S. F., & Neuman, C. *Smart Grid Security*.

[3] Gupta, A., "Consumer Adoption Challenges To The Smart Grid," *Journal of Service Science*, vol. 5, no. 2, pp. 79–86, 2012.

[4] Hess, D. J., & Coley, J. S., "Wireless smart meters and public acceptance: The environment, limited choices, and precautionary politics," *Public Understanding of Science*, vol. 0, no. 0, pp. 1–15, 2012.

[5] Jasso, G., "Factorial Survey Methods for Studying Beliefs and Judgments," *Sociological Methods & Research*, vol. 34, no. 3, pp. 334–423, 2006.

[6] Kostyk, T., & Herkert, J., "Societal implications of the emerging smart grid," *Communications of the ACM*, vol. 55, no. 11, p. 34, 2012.

[7] McDaniel, P., & McLaughlin, S., "Security and Privacy Challenges in the Smart Grid," *IEEE Security Privacy Magazine*, vol. 7, no. 3, pp. 75–77, 2009.

[8] McKenna, E., Richardson, I., & Thomson, M., "Smart meter data: Balancing consumer privacy concerns with legitimate applications," *Energy Policy*, vol. 41, pp. 807–814, 2012.

[9] Potter, C. W., Archambault, A., & Westrick, K., "Building a smarter smart grid through better renewable energy information," *2009 IEEE/PES Power Systems Conference and Exposition*, pp. 1–5, 2009.

[10] Rossi, P. H., & Anderson, A. B., "The Factorial Survey Approach: An Introduction," in *Measuring Social Judgments*, P. H. Rossi & S. L. Nock, Eds., Beverly Hills, CA: Sage Publications, 1982, pp. 15–67.

[11] Watson, R. T., Boudreau, M.-C., & Chen, A. J., "Information Systems and Environmentally Sustainable Development: Energy Informatics and New Directions for the IS Community," *MIS Quarterly*, vol. 34, no. 1, pp. 23–38, 2010.

Invited Talk: Challenges in Security / Security Research

H. Raghav Rao, SUNY Distinguished Service Professor
MSS, SOM, University at Buffalo, State University of New York

THIS TALK will briefly touch upon various challenges in the area of Security and Security Research. It principally touches upon emerging themes in information security that includes tailored trustworthy spaces, moving target, designed-in security, cyber economics and incentives. It also skims over some future challenges in security, in particular the new Science of Security initiative. The talk primarily draws upon two documents: Trustworthy Cyberspace: Strategic Plan for the Federal Cybersecurity Research And Development Program", Executive Office of the President, National Science and Technology Council December 2011 and Federal Cybersecurity Research and Development Program: Strategic Plan, May 2011.

Security Analysis of Certified Wireless Universal Serial Bus Protocol

Rishabh Dudheria and Wade Trappe

***Abstract*— The Certified Wireless Universal Serial Bus (USB) protocol is designed to leverage the existing USB infrastructure in the wireless environment to provide high-speed communication over short distances. We analyze this protocol with a concentration on its security features and identify a number of shortcomings and attacks on the protocol. We have also proposed enhancements to the design of the current protocol to improve its security.**

***Index Terms*—Wireless University Serial Bus protocol, Security analysis, USB**

I. Introduction

THE WIRELESS Universal Serial Bus (USB) protocol [1] is designed to provide robust high-speed wireless connectivity by utilizing the WiMedia MB-OFDM Ultra-wideband (UWB) radio platform [2], [3]. It aims to deliver speeds of 480 Mbps at 3 meters to 110 Mbps at 10 meters.

Security mechanisms for the Wireless USB protocol are implemented on top of the USB core specification. The USB control channel is used to handle all the security operations to make the specification media independent. The goal of Wireless USB security is to provide the same level of protection as the Wired USB 2.0 [4] standard.

The Wireless USB protocol allows each host to connect to a maximum of 127 devices. A group consisting of a Wireless USB host and its associated devices is referred to as a Wireless USB cluster. The Wireless USB standard aims to keep the data exchanged between the host and device private and protected in the wireless environment, along with protecting the owners/user's equipment from an attacker. In order to meet these goals, it provides mutual authentication between the host and the devices during connection setup so that the participating entities have the opportunity to validate each other.

In this paper, we analyze the Wireless USB protocol with an emphasis on its security, list possible attacks, enumerate the points that lack clarity and suggest changes to enhance the robustness of the protocol.

The rest of this paper is organized as follows. Section II provides a brief overview of the security features of the Wireless USB protocol (a detailed description of the protocol is beyond the scope of this paper). Section III lists a collection of possible attacks on the protocol. Section IV lists additional shortcomings of the protocol. Section V proposes changes to the protocol to make it more robust and secure. Finally, Section VI provides a concluding discussion.

R. Dudheria and W. Trappe are with the Electrical and Computer Engineering Department and Wireless Information Network Laboratory at Rutgers University, North Brunswick, NJ 08902 USA (email: {rishabh, trappe}@winlab.rutgers.edu).

Pad bits	Tail bits	FCS	Payload	Tail bits	HCS	MAC header	PHY header	PLCP preamble

Fig. 1. Wireless USB packet format

MIC (8)	Application Payload	WUSB Hdr (2)	SFN (6)	Encrypt Offset (2)	Rsrvd (1)	TKID (3)

Fig. 2. Modified payload for encrypted packet.

II. Overview Of Wireless USB Protocol

A. Packet Format

The packet formats used in the Wireless USB protocol are defined in the corresponding Medium Access Control (MAC) layer standard [3]. The general packet format used in the protocol is shown in Fig. 1. It consists of a physical layer (PHY) preamble, PHY header, MAC header and a data payload. An encrypted packet consists of a modified payload as shown in Fig. 2. The security related fields in the packet are Temporal Key ID (TKID), Reserved (Rsrvd), Encryption Offset, Secure Frame Number (SFN), and Message Integrity Code (MIC). These fields are present in a packet only if the Security bit component of the Frame Control field in the MAC header is set to 1.

There are four basic packet types used in this protocol:

1) Micro-scheduled Management Command (MMC) packet: These are cluster broadcast control packets transmitted by the host using secure packet encapsulation with the Encryption Offset field in the Security Header set to the length of the MMC payload. The host uses the Group Key (that is shared by all the members of the corresponding cluster) to generate the MIC for these packets, which provides authentication of the packet. In particular, the MMC contain Information Elements (IE), which are a part of the information and control mechanisms for the Wireless USB channel.

2) Protocol data packet: This is transmitted either by the host or a device using secure packet encapsulation in most

cases such that the entire body of the application payload is encrypted.

3) Protocol handshake packet: These packets can only be transmitted by the device to the host. When secure packet encapsulation is present, the entire packet is sent in plaintext with its integrity being protected by the MIC.

4) Device notification packet: These are sent by the device to the host mostly using secure packet encapsulation in such a way that the entire packet is transmitted in plaintext with its integrity being protected by the MIC.

B. Encryption

The standard specifies the use of AES-128 [5] Counter mode with CBC-MAC (CCM) as the symmetric encryption algorithm to provide integrity and encryption. The CCM nonce provides uniqueness to each message. The MIC is calculated using the counter-mode blocks. The message and the MIC are encrypted using the keystream provided by the encryption blocks.

C. Keys

The specification defines two types of keys, namely the Master Key and Session Keys.

1) Master Key: It is the long lived key, which is used as a shared secret for authentication and derivation of session keys.

a) Connection Key (CK): The CK is the primary 128- bit key used for establishing connections. Each device and host share a unique CK, which is setup during the initial connection.

2) Session Keys: These are the short lived keys used for encryption and decryption. These keys last only during the lifetime of a connection, i.e., they are discarded when a connection ends.

a) Pair-wise Temporal Key (PTK): The PTK is a 128- bit key used for the encryption and decryption of data packets between the host and the device. It is derived during a 4-way handshake process (refer to Fig. 4). Again, each device should have a unique PTK.

b) Group Temporal Key (GTK): The GTK is a temporal key shared by all the members of the Wireless USB cluster. The host uses the GTK to send secured broadcast messages to all the devices in its cluster. The corresponding devices use the GTK to verify the authenticity (by verifying the MIC) of the broadcast messages received from the host. The standard specifies that the devices may not use the GTK for encryption. However, the standard imposes no requirements on GTK generation.

c) Key Confirmation Key (KCK): The KCK is a 128-bit key used for providing message integrity during authentication. It is derived during authentication and discarded upon completion of authentication.

d) Key Derivation Key (KDK): A KDK (256-bit key) is computed if an additional key is required by any other applications.

Temporal Key ID (TKID): The TKID is a 3 byte ID, which is used by the MAC layer as the name for the GTKs and PTKs. The key used to encrypt a secured packet is identified using the TKID. The TKID values are created by the host and given to the devices at the time of key derivation or key distribution. The specification does not mention how these ID's should be generated by the host.

D. Secure Relationship

The protocol defines a Connection Context (CC) as a secure relationship between a host and a device. The CC needs to be stored in the non-volatile memory at both the host and device end for future reconnections. It consists of 3 pieces of information.

1) Connection Host ID (CHID): The CHID is a unique 128-bit host ID. The CHID is used by the device to locate the host.

2) Connection Device ID (CDID): The CDID is a unique 128-bit device ID, which is assigned by the host to a device during the connection process.

3) CK: The CK is a 128-bit key used to establish connections using this context. This key can be periodically updated by the host.

E. Association Models

Association is defined as the process of establishing the first time connection between the hosts and the devices. The specification mentions four different ways to perform association viz., 'Cable association', 'Numeric association', 'Fixed PIN association', and 'Near Field Communication (NFC) association'. We were unable to find a detailed description of the Fixed PIN association and Near Field Communication (NFC) association models. We hereby provide a short description of the Cable association model and the Numeric association model based on [6]. The devices with a wired USB interface are prescribed to implement the Cable association model, while the devices with a display must implement the Numeric association model. The hosts are supposed to support as many association models as possible.

1) Cable association model: In this model, the user connects the device to the host using a USB cable. The CC is then delivered by the host to the device. The host and the device then perform a 4-way handshake for mutual authentication and to derive the session keys.

2) Numeric association model: In this model, the CK is derived at both ends using Diffie-Hellman (DH) Key exchange [7]. The various steps in the exchange have been shown in Fig. 3 at an abstract level with only the information necessary to understand the model. D represents the device and H represents the host; PK_D and PK_H represent the device and the host public key respectively; SHA-256 is the Secure Hash Standard [8]; DHKey is the secret Diffie-Hellman key; N_D is the number of digits that the device can display; V is the verification number; *Trun-128{}* denotes truncation of all, but the first 128 bits of the argument; and HMAC-SHA-256

represents the Keyed-Hash Message Authentication Code [9] that uses SHA-256 as the hash function.

The device and host each generate a random secret of 256 bits, say A and B respectively. This is used to compute the Diffie-Hellman public key (of size 384 byte), $PK_D = g^A \pmod p$ and $PK_H = g^B \pmod p$. The device then sends a hash commitment of its public key concatenated with the number of digits it can display to the host. This is done to avoid man- in-the-middle attacks. The host then sends its public key to the device. Consequently, the device returns its public key and the number of digits it can display. The host computes the hash of the device's public key concatenated with the number of digits it can display and verifies it with the earlier hash commitment to ensure that the values match; otherwise the association is aborted. After this step, the host and the device independently compute the DHKey. To further protect against a man-in-the-middle attack, the protocol requires that both the host and the device independently calculate a verification number and display 2 to 4 digits of the same so that the user can validate that the association has been completed successfully. Once this is done, both the sides compute the CK. The host then delivers the CHID and CDID to the device. A KDK is then computed independently at both the ends if a key is required by any other application. Finally, all the temporary values computed or generated during this process are erased except the CC and the KDK (if any).

1. D→H: SHA-256 ($PK_D \| N_D$)
2. H→D: PK_H
3. D→H: PK_D and N_D
4. H computes SHA-256 ($PK_D \| N_D$)
5. D computes DHKey = SHA-256 ($PK_H^A \bmod p$)
6. H computes DHKey = SHA-256 ($PK_D^B \bmod p$)
7. Both H and D compute
   V= SHA-256 ($PK_D \| PK_H \|$"displayed digest")
8. Both H and D compute
   CK= Trun-128{HMAC-SHA-256$_{DHKey}$("connection key")}
9. H→D: CHID, CDID
10. If needed, both H and D compute
    KDK= HMAC-SHA-256$_{DHKey}$("key derivation key")

Fig. 3. Numeric association model.

F. 4-Way Handshake

This serves the dual purpose of mutual authentication and session key derivation. The messages exchanged during this process are shown at an abstract level in Fig. 4 with only the information necessary to understand the process.

$H_{N\ once}$ and $D_{N\ once}$ are 128-bit random nonce generated by the host and device respectively; $MIC_{KCK()}$ represents the MIC calculated for the contents inside the bracket with fresh Key Confirmation Key (KCK). New keys are derived from the shared secret CK through a Pseudo Random Function with output length X (PRF-X) as follows:

Key Stream = PRF-256 (CK, Host DevAddr, Device DevAddr, TKID, "Pairwise keys", $H_{N\ once} \| D_{N\ once}$, 32), which is split to form the initial management and data keys. The least significant 16 bytes and the most significant 16 bytes of the Key Stream become the KCK (which is discarded after authentication) and PTK respectively.

The host initiates the 4-way handshake by sending a TKID and H_N once to the device during phase 1. The device then uses this information to generate the KCK and the PTK. During phase 2, the device provides the host with the TKID, D_N once and a MIC calculated over the entire packet payload using the KCK. The host then derives the corresponding keys and verifies the MIC provided by the device. Upon successful verification, the host has proof that the device holds the correct CK. Consequently, it proceeds to phase 3 to provide proof to the device that it holds the correct CK by sending a message containing the TKID, $H_{N\ once}$ and a MIC computed over the entire packet payload with the KCK. The device computes the corresponding MIC and verifies it with the MIC sent by the host to ensure that the values match, stores the session key, and informs the host that it has successfully installed the session key during phase 4; otherwise it aborts the connection and removes the derived session key.

Phase 1
H→D: TKID, H_{Nonce}
Phase 2
D→H: TKID, D_{Nonce} , MIC_{KCK}(TKID,CDID, D_{Nonce})
Phase 3
H→D: TKID, H_{Nonce} , MIC_{KCK}(TKID,CDID, H_{Nonce})
Phase 4
D→H: Successfully installed session key

Fig. 4. 4-way handshake.

After the successful completion of the 4-way handshake, the host provides the device with the current GTK, which is sent in a secured manner using the PTK.

G. Replay Prevention

The protocol defines three 48 bit components: a Secure Frame Counter (SFC), SFN and a replay counter (RC) to avoid replay of packets sent using the GTK and the PTK. The host and the device each maintain a separate SFC to encrypt transmitted packets, and a separate RC to decrypt received packets for each session key. Initially, these counters are set to a value of zero when a new key is installed. The SFN is an image of the SFC being used to encrypt the packet.

The SFC associated with the key is incremented when a packet is encrypted for transmission and is then copied to the SFN field of the packet. The SFN value for each successive retry of a packet is greater than the last attempt.

The receiver compares the packet SFN value with the value of the RC associated with the decryption key. A packet is declared to be a replay of a previous packet if the SFN value is less than or equal to the value of RC (and is consequently discarded); otherwise the RC is set to be equal to the SFN of the received packet.

A. Key Management

Hosts are responsible for all the key management

operations. The PTK can expire if it is used to encrypt 2^{48} packets. In such a case, the host performs an additional 4-way handshake with the device to derive a new PTK. The GTK is changed whenever a device leaves the Wireless USB cluster. Since, synchronization of the new GTK is difficult, the protocol requires each device to be capable of holding two GTKs. The host distributes the GTKs in numerical order based on a 4-bit index value. The low bit of the session key index is used as the table index so that Key_2 can replace Key_0, Key_3 can replace Key_1 , and so on. The host installs and begins to use a new GTK only after the last device in the cluster has confirmed receiving it. The device can discard an older GTK only after the host begins to use the new GTK. The host must not use the older GTK once it starts using the new GTK. However, the specification does not mention what should be done if the same GTK is used to encrypt more than 2^{48} packets.

III. Threat Analysis of Wireless USB Protocol

We now look at several attacks that may be conducted against the Wireless USB protocol. To begin with, we assume an attack model in which an attacker can eavesdrop on every message, replay stored messages, and insert forged messages; but it does not know the shared CK and PTK between the host and an uncompromised device. We further assume that it is also possible for an attacker to intercept and block delivery of a message. This assumption is further supported by the fact that the messages tend to experience higher loss rates in a wireless medium.

A. Message Spoofing

A device that has successfully established a connection with the host and has acquired the GTK can masquerade as the host and send forged MMC packets to the cluster containing the following Information Elements (IEs), which may lead to a denial of service (DoS) attack:

1) WHOST DISCONNECT IE: To inform all the cluster members that they are being disconnected resulting in spoofed disassociation message.

2) WCHANNEL STOP IE: To notify cluster members that the Wireless USB channel is being stopped.

3) WCHCHANGEANNOUNCE IE: To notify cluster members that the host is moving the Wireless USB channel to a different PHY channel.

B. Sybil Attack

A successful connection between a device and a host requires the CC (consisting of 16 bytes of CHID, CHID, and CK each) to be stored in the non-volatile memory at both the device and host end. A malicious device can make 127 connections to the same host if it has enough memory ($16 \times 3 \times 127$ bytes ≈ 6 KB). This can exhaust all the resources of the host and thus, prevent any legitimate devices from accessing the corresponding host leading to a Sybil attack [10].

C. Water Torture Attack

In order to save power, the host can go in sleep state, while enabling the devices to perform remote wakeup. A malicious device can prevent the host from sleeping, effectively draining off its power.

D. Packet Dropping

An adversary can change the SFN associated with the group key in a packet from its original value to a much higher value, and then recalculate the MIC associated with the corresponding packet. This causes legitimate packets with lower SFN values to be dropped by the devices.

IV. Short Comings/Lack of Clarity

The protocol does not impose any requirements on CK generation in the Cable association model, and when new CKs are distributed by the host using the 'Set Connection Context' command. The specification requires the host to distribute a unique CK to every device. This means that the CK should be generated randomly using a uniform probability distribution on the space of 128-bit strings. The standard should specify this explicitly.

The protocol also fails to define how the GTK and the TKID are generated by the host. We were also unable to locate the size of the GTK in the specification. Furthermore, the standard also fails to mention how the KDK would be generated by the host and the device if the initial association were to take place using the Cable association model.

The specification defines the CCM nonce to be made up of the SFN, TKID, DestAddr and SrcAddr to ensure its uniqueness. Now, if there are more than 2^{48} packets encrypted with the same GTK, then the SFN associated with the corresponding packets would roll over resulting in the same nonce to be used again for encryption. The standard should clearly specify what should be done if the same GTK is used to encrypt more than 2^{48} packets.

The standard specifies that the Set Connection Context command can be used by the host to modify a device's CC. It also mentions that the CC delivered in this way should always be protected. The designer's should have explicitly mentioned that the CC's delivery has to be protected using the PTK; otherwise, a malicious device can announce a change in the physical channel being used by the host and then use this command to achieve a session hijack.

Chapter 7, p. 163 of the specification mentions about some host association keys that are distributed by the host to the devices using the 'Set Key' command. We could not comprehend what these keys referred to as they were clearly not PTKs or GTKs.

The latest version of the standard [1] refers to an 'Association Model Specification' document on p. 141 and 142 of chapter 6, which we were unable to find on the web. The previous version of the Wireless USB specification [11] has a corresponding 'Association Models Supplement' [6] document that was published on March 2, 2006. However, [1] seems to be referring to some other document as it describes a new Fixed PIN association model on p. 142 of chapter 6,

which has not been described in [6].

V. ENHANCEMENTS

A. A. Elliptic Curve Diffie-Hellman (ECDH) Exchange

The designers have used 3072 bit ephemeral DH key for initial connection in the Numeric association model, which provides a cryptographic strength of 128 bits. We are proposing the use of ephemeral ECDH keys for computing CK in the Numeric association model as it can provide the same cryptographic strength with a lower key size of 283 bits. This smaller key size can save a lot or resources such as memory, computing power and battery [12]. 'Association Models Supplement FAQ' [13] mentions that the Menezes-Qu-Vanstone (MQV) cryptographic method was considered for Numeric association, but was rejected because it is covered by patents and requires excessive computation. ECDH on the other hand is non-proprietary. There are several standards available from IEEE, IETF, etc., providing guidelines regarding its practical deployment.

A brief review of ECDH is provided here as a detailed description of this algorithm is beyond the scope of this paper. The curve parameters are fixed in advance for all the hosts and devices designed to use the Wireless USB protocol. Initially, both the device and host create their private keys by generating a random integer. They calculate their public key by multiplying their private key with the generating point. Then, they exchange their public keys over the wireless medium. The shared secret is generated at each end by multiplying the public key received from the other end with the local private key. Let the device and host generate the random private keys A and B respectively. The device and host then calculate their public keys X and Y as follows: X = A.G and Y = B.G, where G is the basepoint on the curve. The public keys X and Y are then exchanged over the wireless channel. The device and host then derive the shared secret by computing A.Y and B.X respectively.

Thus, the same steps as described in the Numeric association model can be implemented by using the ephemeral ECDH keys instead of the ephemeral DH keys.

B. CK Generation

Another source of weakness in the current protocol is that the host generates the CK in the Cable association model as well as when new CKs are distributed using the Set Connection Context command on its own. This principle implies that a device must trust that the host always generates a new CK that is cryptographically separated from all the other CKs generated by all the hosts. Further, if the CKs are generated by the host using a random number generator then it must be perfect; otherwise, if it exhibits some bias then it can expose the CK and hence the PTK. A safer way to compute the CK would be to calculate it using the bits contributed by both the host and the device; for example, CK = HMAC-SHA (host generated connection key, some random value generated by the device). This is similar to the weakness associated with the generation of authorization key in 802.16 [14].

VI. DISCUSSION

The Wireless USB protocol employs information that is sent in the clear, but what saves the protocol from an outsider attack is the use of out-of-band mechanisms (USB cable, user) to validate the association. For instance, in the Numeric association model any entity can capture the public keys and hence display the correct digits required for verification, but it still cannot cause any harm to the participants as the user needs to validate the association. Our paper has pointed out many shortcomings in the current specification of the Wireless USB protocol. These should be fixed by the designers as soon as possible to avoid ambiguity once the products enter the market. We have also outlined a number of ways to enhance the security of the current protocol: the most important of them being the use of ephemeral ECDH keys instead of ephemeral DH keys for initial association.

REFERENCES

[1] Wireless Universal Serial Bus Specification, Wireless USB Promoter Group Std., Rev. 1.1, Sep. 2010. [Online]. Available: http://www.usb.org/developers/wusb/docs

[2] Multiband OFDM Physical Layer Specification, WiMedia Alliance Std., Rev. 1.2, Feb. 2007.

[3] Distributed Medium Access Control (MAC) for Wireless Networks, WiMedia Alliance Std., Rev. 1.2, Mar. 2008.

[4] Universal Serial Bus Specification, Universal Serial Bus Implementers Forum (USBIF) Std., Rev. 2.0, Apr. 2000, including all published errata. [Online]. Available: http://www.usb.org/developers/docs/

[5] "FIPS PUB 197, Advanced Encryption Standard," National Institute of Standards and Technology, Nov. 2001. [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[6] Association Models Supplement to the Certified Wireless Universal Serial Bus Specification, Wireless USB Promoter Group Std., Rev. 1.0, Mar. 2006. [Online]. Available: http://read.pudn.com/downloads162/sourcecode/unix linux/network/736567/wusb 2007 0214/WUSB AM Spec r10.pdf

[7] E. Rescoria, "Diffie-Hellman Key Agreement Method," RFC 2631, Jun. 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2631.txt

[8] "FIPS PUB 180-2, Secure Hash Standard," National Institute of Standards and Technology, Aug. 2002. [Online]. Available: http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf

[9] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC:keyed-hashing for message authentication," RFC 2104, Feb. 1997. [Online]. Available: http://www.faqs.org/rfcs/rfc2104.html

[10] J. R. Douceur, "The Sybil Attack," in Revised Papers from the First International Workshop on Peer-to-Peer Systems, ser. IPTPS '01. London, UK, UK: Springer-Verlag, 2002, pp. 251–260.

[11] Wireless Universal Serial Bus Specification, Wireless USB Promoter Group Std., Rev. 1.0, May 2005. [Online]. Available: http://www.usb.org/wusb/docs/WirelessUSBSpecification r10.pdf

[12] K. Lauter, "The advantages of elliptic curve cryptography for wireless security," Wireless Communications, IEEE, vol. 11, no. 1, pp. 62–67, Feb. 2004.

[13] "Association Models Supplement to the Certified Wireless Universal Serial Bus Specification Frequently Asked Questions," Wireless USB Promoter Group, Jun. 2007. [Online]. Available: http://www.usb.org/developers/wusb/WUSB AM FAQ 2007 06 19.pdf

[14] D. Johnston and J. Walker, "Overview of IEEE 802.16 security," Security Privacy, IEEE, vol. 2, no. 3, pp. 40–48, May-June 2004.

Malware Analysis for Android Operating

Kriti Sharma, Trushank Dand, Tae Oh and William Stackpole

***Abstract*—The number of mobile devices has dramatically increased in the last decade. As the mobile devices become more pervasive, users increasingly use it more frequently for everything since they are small, portable, and easy to use. Users store all their sensitive data and information on these devices. However, this ease of use comes at a very big price and comes with side-effects which most users are unaware of. With the increase use of mobile devices, malware is also enjoying unprecedented growth at the expense of unsuspecting and naïve users. Even though several mobile security solutions have been proposed, it is apparent that more effort is required to ensure the security of the data on these devices. The research presented in this paper is an attempt to analyze malware behavior by combining code review and live testing to collect and analyze data in an effort to suggest security techniques not currently found in Android-based devices. The results of this research will provide and insight into the targets and actions of malware as well as provide higher security if the techniques are coded into the Android OS.**

***Index Terms*—Mobile security, Malware, Android, Anti-Virus**

I. Introduction

SINCE THE launch of Google's Android OS in 2008, the smartphone market grew stupendously and has never looked back. As of October 2012, according to estimates, out of the total number of smartphones shipped, seventy-five percent were Android devices [6]. Google claims that each day there are approximately 1.3 million activations and the total number of Android devices exceeds 5 million making it the most widely used mobile operating system today [5]. Due to its ever-increasing numbers and easy acceptance into Google's App Store, the malware market is also thriving and enjoying unprecedented growth. To counteract these threats, anti-virus and anti-malware companies are making considerable efforts. However, these efforts fail to keep up with the current security requirements. Thus, this research makes an effort to try and develop additional mitigation techniques, which will be more effective in handling the threats.

This paper begins with the introduction of the Android OS and follows with a description of the tools used to perform code review and analyze the behavior of the malicious apps. Then the paper leads the previous work in the field and the research methodology used for analysis. Then, the results and the description of the data will be discussed and the paper will conclude with the data interpretation.

K. Sharma is with the Department of Information Sciences and Technologies (IST), Golisano College of Computing and Information Sciences, Rochester Institute of Technology, Rochester NY 14623.
T. Dand is with the Department of Computer Science, Golisano College of Computing and Information Sciences, Rochester Institute of Technology, Rochester NY 14623 (email: tgd2900@rit.edu).
T. Oh is with the Department of Information Sciences and Technologies (IST), Golisano College of Computing and Information Sciences, Rochester Institute of Technology, Rochester NY 14623 (email: Tom.Oh@rit.edu).
W. Stackpole is with the Department of Computing Security, Golisano College of Computing and Information Sciences, Rochester Institute of Technology, Rochester NY 14623 (email: Bill.Stackpole@rit.edu).

II. Android and Need for Analysis

A. The Android Operating System

Google's Android OS is built on the ARM platform with a modified Linux kernel of version 2.6.x (versions older than 4.0) or 3.x (version 4.0+) and was released in 2008. Lower level system utilities are written in C while most user "applications" are written in Java, although it is possible to write applications using native code in a language such as C++ [7], [8]. Google's custom Dalvik virtual machine is a replacement for the standard Java virtual machine used on other desktop and server platforms; the engine is optimized for limited resources typically available on mobile devices. Therefore, if a developer writes an Android application in Java, it will be compiled into dex bytecode (in a file called classes.dex), not standard Java bytecode that would run on Windows or UNIX platforms. A detailed presentation of Dalvik can be found in Dalvik author Dan Borstein's presentation. [20]

The most interesting feature about Android is that the kernel places each application in a sandbox when it executes. This isolates the application from all the other applications and other parts of the operating system. This involves the use of standard UNIX process separation techniques which allow the application to access its stored data and memory without being able to interfere with the other applications hardware, memory and data usage. Each application is assigned a unique UID (user ID) and GID (group ID). User can install their choice of applications from the Google Play Store or they can directly install them in the memory card. While installing the applications, the user is presented with certain permissions requested by the application like access to the Internet, access to GPS coordinates, accessing contacts, etc. The user can either choose to accept all permissions requested by the application or choose to not install the application.

B. Need for App Analysis

Since their initial introduction to this world, mobile devices have seen considerable innovation and creativity in terms of their features and functions. From devices that once were only used to make calls and send texts, mobile phones can now present users with calendars, web browsers, task managers, games, and email access, among other features, resembling desktop computers in terms functionality. This increasing complexity of smartphones brings with it increasing vulnerabilities. Users entrust more and more sensitive data like

banking data, social networking identification to the security mechanisms embedded within these mobile devices and operating systems. It is apparent that the current security technologies are insufficient and there is a need to assess the Android OS and application software for malicious activity.

C. Existent Security Issues

Applications in Android devices may be installed through either the Google Play Store (formerly called Android Market) or through a variety of third party application stores. Some of the third party stores are Opera Mobile app store, GetJar, SlideME, etc. The fact that Android permits application installation from third party vendors means that Google has no control over the quality or safety of the applications provided in these stores. Several cases were encountered where legitimate apps from the Google Play Store were modified to inject malicious code and the modified apps were sold in these third party stores. It is difficult to determine whether the application is genuine or not. In these cases, the reliability of the application depends upon the security measures implemented by the application store. This makes it essential to provide a reliable means to verify the authenticity of the applications.

D. Currently Available Security Measures

Google's Play Store has a security enforcement known as the "Bouncer" which verifies the applications being uploaded for any suspicious behavior. It works on a blacklisting based approach and instances have occurred where-in the malicious application survived several hours or even days before it was detected and taken off. However, this app still does not protect the users from installing malicious apps from sources other than the Play Store. 4.2 version (Jelly Bean) of Android has shipped with a security feature to counter this problem. A new feature allows the user to verify the third party application being installed on the phone. However, research published by Jiang [21] suggests that the system has a malware detection rate of 15.32%. An anti-malware system must have an detection rate of at least 80% to be deemed acceptable.

III. Past and Present Mobile Application Analysis

The earliest research on the current generation of mobile devices dates back to 2007, when iOS, Blackberry and Android operating systems began to appear in the market. When security became a major concern, many researches followed. Cheng et al. developed SmartSiren: an intrusion detection system for the Symbian OS and Windows Mobile [3]. A monitoring agent runs on the mobile device and it collects and records the system calls made by the applications running on the phone. This data is forwarded to the researcher's server over the Internet. Each device registers with the proxy and provide information so that the proxy can differentiate between the data obtained from the different devices. This data is then analyzed to find out if the device is infected with some malware that uses either SMS or Bluetooth to spread. Bose et al. also proposed a behavior-based malware detection system for Symbian OS in 2008 [1]. Known families of existing malware are used to collect data regarding system events and API calls and a logical flow diagram is constructed and captured as a signature for that family. Then, any new system or application is compared against this database of signatures. If a match is found, the application is considered malicious.

pBMDS was another system proposed by Xie et al. that compares inputs to applications instead of reading and creating logs of malware behavior [2]. The basis of this system was that user input differs from the automated input. This difference was then used to identify abnormal behavior that may result from malware on the device.

Another monitoring system was developed by Houmansadr [4]. This system was cloud-based in which an emulated device would run in the cloud. The mobile device sends all its network traffic through a controlled proxy, which duplicates this device traffic on the emulator running on the cloud. A monitoring system connected to the emulator would analyze this traffic and would alert the monitoring agent running on the mobile device regarding the malicious activity to take protective action.

All the above-mentioned studies suggest that mobile security is increasingly incorporating real-time monitoring. However, it is apparent that none of these studies incorporated code review. A major component if this research is to analyze the source code to collect data, which will be a significant step in the field of malware analysis and provides critical information.

IV. Methodology

The methodology to be followed for reviewing applications has two phases: 1) code review and 2) live testing. First, malware samples are retrieved from security research repositories and quarantined and sorted into categories like Trojan, worm, spyware, etc. This classification helps in the analysis by categorizing malware, which may share similar behavior. Then we proceed to analyze the source code of the apk files to determine coding style, encryption, obfuscation and infer the behavior that the malware is expected to exhibit once installed in the device. In the next phase, the malware samples are run on the device emulator to observe the malware interaction with the device and the user. The results of both these phases are then compiled, interpreted and compared to other samples belonging to the same family. Initially, these phases are carried out manually to verify effectiveness. Both the phases will be automated in the future to minimize human input. To complete the investigation of the malware samples, the researchers used the following tools:

TABLE I
MALWARE INVESTIGATION TOOLS

Tool Name	Purpose	Process Used
JD-GUI	Java disassembly and analysis	Code Review
Dex2Jar	Dex to Java bytecode translation	Code Review
Axmlprinter	Binary XML Translation	Code Review
Apktool	Android package (apk) management	Code Review
Androgiard	Apk live reverse engineering	Code Review
Smali/Baksmali	Dex assembler/disassembler	Code Review
Android Emulator	Android device emulation	Live Testing
Android Dalvik Debug Monitor	Remote device monitoring	Live Testing

Server (DDMS)		
IDA Pro	ARM and Dex disassembly	Code Review/Live Testing

The use of both phases compliments each other as malware will be detected in the code review and obfuscated code will be detected in the live testing.

A. Code Review

This phase involves analyzing the source apk file and all its contents. The contents are made accessible by converting them into an interpretable form. The classes are converted from DEX to Java classes, and the binary XML file, androidmanifest.XML, is converted to readable XML. Once the conversion is successful, the code is reviewed for any suspicious behavior. This is done by going through the required permissions of the application in the actual code, etc. A log is then made for all behaviors noticed that may be deemed malicious.

The following tools are required for code review:

- DEX2JAR
- AXMLPrinter2
- Java Decompiler

The code review begins with converting the DEX code to Java using DEX2JAR. The Java shows all the resources and class files that are used for the app. Then, the Java Decompiler converts the class files into readable format. Also, the binary XML is converted to readable format by using AXMLPrinter2.

Once all the content is ready, the actual code is analyzed. The reviewer goes through the androidmanifest.XML file and extracts the permissions requested by the app. The activities within the app are also noted. The next stage is to review the source code extracted previously. Reviews look out for specific API calls like those to SMSManager classes to find specific behavior (sending SMS in this case). A report is generated for all of these instances.

B. Live Testing

In this phase, the apk file is actually installed on an emulated device and the behavior is observed. A report is generated and behavior is compared with the expected results from the code review.

Live tests are performed by running the app on actual devices and emulators. Various metrics, such as, process threads, network flow, and file access are monitored. Nexus S is used for live testing, as it is a vanilla Android OS without any OEM software. This gives a more generalized view to the tests, which makes it applicable to a variety of devices. Tcpdump is used for capturing network traffic, and DDMS is used to analyze process threads and other metrics.

A live testing tool is developed in order to assist in installing the application, accessing the shell and capturing metrics. This takes the effort of typing tedious commands in a terminal off the tester and hence improves efficiency.

The app is installed using the tool, and the network capture is started. The application is started and used as intended. Report from the code review is referred to possibly generate the conditions required for the malware to become active. The packet capture and other metrics captured are analyzed and a report is generated.

V. Aggregation of Findings

Once both the code review and live test phases are done, the findings are aggregated and a final report is generated which contains information about the category of the malware, its behavior, resources it targets, etc.

A. Code Review

Samples from over 50 family groups of Android malware were analyzed and they were classified into the following categories (with duplicates):

- Spyware: 17
- Trojans: 33
- SMS: 32

The category SMS is more of a meta category and these apps can be counted in other categories. However, these malware are unique to the mobile platform and are thus discussed separately. The common behaviors found in malware based on the permissions requested and source code analysis are presented in Table 2.

TABLE II
COMMON OBSERVED MALWARE BEHAVIOR

Common Behaviors	Count
Receives SMS/MMS	25
Sends SMS/MMS	25
Send Data over HTTP(s)	23
Uses WiFi	20
Write to disk (internal or external flash card)	20
Obfuscation	20
Send Data(cellular)	19
Receive Data over HTTP (s)	18
Access Device Location	18
Receive Data (cellular)	15
Reads from Disk (internal or external flash card)	14
Can execute commands	12
Mount/ Unmount Filesystems	11
Encryption	6
Set Network Properties	6
Send Data (Raw)	5
Receive Data (Raw)	4

As mentioned above, SMS related activity is the most commonly observed behavior. Most apps request permission to send/receive SMS, and most of these are from premium numbers or command and control servers. However, it is important to note that not all SMS malware sent and received SMS. Some of malware only requested permission to send SMS and some only requested to receive SMS. The other most common behavior was to send/ receive data using a http connection. Again this was done to contact some command and control server and to transfer the data from the device to the server. Obfuscation was also observed in most of the malware samples within the source code as most of the class names and variable names were changed to single letters. This though does not impact how the app interacts with the device, it makes the app difficult to reverse engineer. Most apps also requested access to location coordinates either through the network or the GPS, and some of the apps also exhibited functionality for remote control.

In addition to these common behaviors, some additional behaviors were also noticed which were not common among the apps. These uncommon behaviors are mentioned in Table 3.

TABLE III
UNCOMMON OBSERVED MALWARE BEHAVIORS

Uncommon Behaviors	Count
Record Calls	3
Record Voice	2
Root Aware	2
Reboot Device	2
Make Calls	2
Take Images	1
Uses Bluetooth	1
Code Reflection	1
Terminate Processes	1

These uncommon behaviors are subversive and may cause many disruptions with the mobile device. Behavior like terminating processes and rebooting the device are usually done to create optimal execution condition for the malware. Some applications also had the functionality to make use of rooted device and request escalated privileges.

The second important metric for malware analysis is the types of data being targeted by malware, detailed in Table 4.

TABLE IV
OBSERVED MALWARE DATA TARGETS

Data Target	Count
SMS/MMS	30
IMEI	19
Phone Number	13
Contacts	11
Email	9
Android Version	9
SDK Version	9
Browser History	9
GPS Coordinates	9
Cellular Carrier	7
Data in Flash card	7
Call Logs	5
Phone Conversations	4
Photos/Videos	3
Root Level	2
Access Point	1

As described in Table 2, SMS and MMS are the most commonly accessed type of data on the device. These are mainly used to send this data obtained from the device to the attacker or the person controlling and distributing the malware. This was done to leverage any exploits against the device.

Two additional items have been observed from the analysis. Many malware samples had requested the RECEIVE_BOOT_COMPLETED permission upon installation. Google identifies this permission as granting access to a system broadcast when the system has booted, and applications can safely start in the background [9]. This effectively allows the sample to know when it can start itself, eliminating the need to require the user to manually start the application each time the device boots. Several samples also established high-priority listeners for specific actions, such as incoming SMS. Listeners are granted access to the type of data requested in descending numerical order, which allows the malware to intercept data retrieved using the corresponding API. In many cases, the malware is granted exclusive access to this data, which can be used to prevent a legitimate application from receiving data that would alert the user to its presence on the device. Parsing the manifest file and finding very high listener values may be a good indicator that the sample is malicious.

B. *Live Testing*

The researchers selected the Nexus S for running the live tests; the main reason for this is that this device comes with stock Android, which provides a general view for the purpose of this research. The live testing is based on the results of the code review and effort is taken to trace the suspected behavior from the code reviews to the live tests. The behavior testing as live testing is commonly called is done on the emulator and the actual Nexus S device. The reason for performing this 2-layered test is twofold. First, it is very difficult to simulate the required test conditions on the emulator. Secondly, some malware may be aware of being run on the emulator.

The testers use DDMS, which provides port-forwarding services, screen capture, thread and heap information, logcat, process, and radio state information, incoming call and SMS spoofing, location data spoofing, and more, to track the data going in and out of the device and also all the processes being carried out by the application in question.

Wireshark was used for traffic capture where it was difficult to capture data using DDMS. The observations made by the researchers clearly depict that the results of the code review and live testing match.

VI. CONCLUSION

In this paper, a method is presented by which Android applications can be assessed for malicious activity in two ways: through code review and live testing. The assessment of known Android malware using this method yields sufficient knowledge to propose several technical solutions for Android that currently do not fully exist. The solutions aim to increase confidentiality, integrity, and availability of the system by improving the isolation of data, protecting access to data, and mitigating the threat of service interruption to data or applications. Thus, it is very apparent that other technical and conceptual solutions must be formulated which are more efficient in thwarting the attempts of the malware developers to access private and sensitive data and exploit the vulnerabilities of the mobile operating system.

VII. FUTURE WORK

The research described in this paper is ongoing. Research will continue in investigating the source code of newly identified Android malware. To improve the quality and efficiency of this process, it is desirable to invest in the design, development, and implementation of a fully

automated application code review system that is employed to triage arbitrary Android applications (with the eventual hope of supporting any number of programming languages and packages). The live testing process will be rapidly expanded on available malware samples using isolated networks with the Android emulator and physical Android devices. The possibility of developing the process intelligence system for Android is also very appealing; the ability to gather process performance data from millions of devices can provide a wealth of information to facilitate all manner of intelligence and defense mechanisms, as discussed earlier in this paper. Devices participating in this system can benefit from the information gathered from all other devices, contribute to the identification of "bad" apps, and aid in the refinement of techniques that can help better protect Android devices and their users.

REFERENCES

[1] Bose, X. Hu, K. G. Shin, and T. Park, "Behavioral detection of malware on mobile handsets," in Proceedings of the 6th international conference on mobile systems, applications, and services (Mobisys '08)., New York, NY, USA: ACM, 2008, pp. 225-238.

[2] L. Xie, X. Zhang, J. Seifert, and S. Zhu, "pBMDS: a behavior-based malware detection system for cellphone devices," in Proceedings of the third ACM conference on wireless network security (WiSec '10)., New York, NY, USA: ACM, 2010, pp. 37-48.

[3] J. Cheng, S. H. Y. Wong, H. Yang, and S. Lu. "SmartSiren: virus detection and alert for smartphones," in Proceedings of the 5th international conference on mobile systems, applications, and services (Mobisys '07)., New York, NY, USA: ACM, 2007, pp. 258-271.

[4] Houmansadr, S. A. Zonouz, and R. Berthier, "A cloud-based intrusion detection and response system for mobile phones," in Dependable systems and networks workshops (DSN-W), 2011 IEEE/FIP 41st international conference on, June 2011, pp. 31-32.

[5] M. Burns, (2012 September 5). Eric Schmidt: "There are now 1.3 million Android device activations per day" [Online]. Available: http://techcrunch.com/2012/09/05/eric-schmidt-there-are-now-1-3-million-android-device-activations-per-day/

[6] IDC, (2012 November 1). Android marks fourth anniversary since launch with 75% market share in third quarter, according to IDC [Online]. Available: http://www.idc.com/getdoc.jsp?containerId=prUS23771812

[7] Ohloh, (2012 November 29). The Android open source project [Online]. Available: http://www.ohloh.net/p/android

[8] Google, (2012 November 1). Android source code [Online]. Available: http://source.android.com/source/downloading.html

[9] Google, (2012 November 4). Android permissions [Online]. Available: http://developer.android.com/reference/android/Manifest.permission.html#RECEIVE_BOOT_COMPLETED

[10] S. Smalley, C. Vance, and W. Salamon., "Implementing SELinux as a Linux security module," NAI Labs Technical report 01-043, 2001.

[11] S. Smalley, (2012 November 23). The Case for SEAndroid [Online]. Available: http://selinuxproject.org/~jmorris/lss2011_slides/caseforseandroid.pdf

[12] SELinux Project, (2012 November 23). SEAndroid - SELinux Wiki [Online]. Available: http://www.selinuxproject.org/page/SEAndroid

[13] J.R. Raphael, (2012 November 1). Inside Android 4.2's powerful new security system [Online]. Available: http://blogs.computerworld.com/android/21259/android-42-security

[14] V. Samar and R. Schemers, (1995 October). RFC 86, Unified Login with Pluggable Authentication Modules (PAM) [Online]. Available: http://www.opengroup.org/rfc/mirror-rfc/rfc86.0.txt

[15] Google, (2012 November 16). Android Debug Bridge [Online]. Available: http://developer.android.com/tools/help/adb.html

[16] R. Amadeo, (2012 October 17). Android 4.2 Alpha Teardown, Part 2: SELinux, VPN Lockdown, and Premium SMS Confirmation [Online]. Available: http://www.androidpolice.com/2012/10/17/exclusive-android-4-2-alpha-teardown-part-2-selinux-vpn-lockdown-and-premium-SMS-confirmation/

[17] Rodrigo ZR, (2012 November 01). DroidWall - Android Firewall [Online]. Available: http://code.google.com/p/droidwall/

[18] R. Ramachandran, T. Oh, and B. Stackpole, "Android Anti-virus Analysis", ASIA and SKM 2012, Albany, NY, June 2012. (Best Techical Paper)

[19] Google (2012 November 30). Android Security Overview [Online]. Available: http://source.android.com/tech/security/index.html#the-application-sandbox

[20] Borstein (2012 November 30). Presentation of Dalvik VM Internals [Online]. Available: http://sites.google.com/site/io/dalvik-vm-internals/2008-05-29-Presentation-Of-Dalvik-VM-Internals.pdf?attredirects=0

[21] X. Jiang, "An Evaluation of the Application Verification Service in Android 4.2," 10-Dec-2012. [Online]. Available: http://www.cs.ncsu.edu/faculty/jiang/appverify/.

Android Malware Analysis Platform

Ben Andrews, Tae Oh and William Stackpole

***Abstracts*—Malware for smartphones is a prominent threat to secu- rity, with Android leading the charge as a primary threat. To combat this, it is vital that the security field devote research into finding better methods of malware analysis and subsequently malware defense. As a result, initiative was taken to perform focused research on this subject in the form of an independent study. The purpose of the study is to develop a lab virtual environment for analyzing Android malware. The solution needs to be intuitive and centrally managed in order to be effective for a professor and their assistants. Work has begun on such a system, and the plan is to continue developing an environment until an adequate solution has been produced.**

***Index Terms*—Android OS, Malware analysis, Mobile device security, Malware**

I. Introduction

TECHNOLOGY CONTINUES to advance at an ever-increasing rate, and the mobile technology field is no exception to this trend. Smartphones have become an extremely popular commodity with 40% of mobile users within the United States own smartphones [3]. It has been seen in the past with traditional computing devices that as popularity increases, threats to their security generally increase as well. This can be exemplified by the prominence of malware for Windows operating systems, which make up over 91% of the operating system market according to a recent study [5]. Most malware writers are financially motivated, and seek targets where they can make the most profit. It makes sense to shape their attacks to affect the largest number of devices possible. Too often history has a way of repeating itself, and this can be seen with the onset of mobile malware. Android phones occupy 40% of the smartphone market [3], and accordingly, malware analysts predict the number of Android malware cases will reach 1 million within the year 2013 [4]. Current solutions for protecting against such threats are not sufficient. The built- in malware scanner implemented by Google into the Jelly Bean Android version only had a 15.32% success rate when detecting malware [2].

B. Andrews is with the Golisano College of Computing and Information Sciences, Rochester Institute of Technology, Rochester NY 14623 (email: bla7168@rit.edu).

T. Oh is with the Department of Information Sciences and Technologies (IST), Golisano College of Computing and Information Sciences, Rochester Institute of Technology, Rochester NY 14623 (email: Tom.Oh@rit.edu).

W. Stackpole is with the Department of Computing Security, Golisano College of Computing and Information Sciences, Rochester Institute of Technology, Rochester NY 14623 (email: Bill.Stackpole@rit.edu).

II. Malware Analysis Environment

To meet the needs of an educational environment, the final solution must have a few different components. There must be a means of distributing the malware and virtual devices, configuration reporting for student progress tracking and configuration, and tools to manage multiple users.

A. Environment Overview

The current environment is comprised of two virtual machines within VMware Workstation running Ubuntu version 12.04 as their operating systems. One of these virtual machines has been designated as a server (operated by professors) and one as a client (used by students). Each virtual machine is loaded with a set of scripts appropriate for its role, which will be discussed later. The server is hosting both malware and virtual Android devices which are distributed to the client through use of an FTP server. Android configurations on the client are pulled from the necessary files and sent to the server through a client/server system written in Python. These configurations are then stored in HTML files corresponding to the specific client and then accessed by an Apache web server. All scripts read from two configuration files to customize the client environment. One file contains the address of the server, and the other contains the name of the user of the client. This user value is set by a python script that runs when a user logins in to their virtual machine. The script displays a GUI prompt that asks for the desired username, and places this into the configuration file. The username must be a valid name of a user on the malware server, as this is what is used for authentication during synchronization and configuration reporting. Android virtual devices are provided by the Android Software Development Kit.

B. Android Software Development Kit

The Android Software Development Kit, provided by Google, is an extensive package that includes methods of virtualizing Android devices for testing and development on multiple operating systems [1]. This virtualization system proves useful for a couple of reasons in respect to malware research. The "snapshot" feature of the SDK virtualization implementation, or emulator, allows for a user to set a device to a certain state and then save this state. The user can then close the device, or perform any action desired to the device, and then simply load the previous state to eliminate any changes. This means that virtual Android devices can be delivered to students in desired states of infection and can be analyzed accordingly. Also, if the student happens to reach an

undesired state at any time during operations, they can simply revert to the included snapshot and start again. This is also useful when testing malware applications that require root access to a device. A typical Android emulator loses root privileges when rebooted. Through use of the snapshot tool, the rooted state can be preserved. Additionally, the SDK provides emulation for a wide range of Android versions. Configurations within the environment currently include both rooted and standard virtual devices for Android Gingerbread, Ice Cream Sandwich, and Jelly Bean, run on a virtual Nexus S. This provides a comprehensive approach for meeting the requirements of simulating any Android malware infection.

C. Emulator and Malware Distribution

To distribute the configured Android emulators and malware to student computers, a system has been constructed that utilizes a VSFTP file server and bash scripts/utilities. On the server, these entities are placed in a designated folder. The FTP service is configured to "chroot" users into their home directory. This just means that each user can only see the files that are located within their home directory. Instead of creating home directories for each user, symbolic links have been created with the same name that their home directory would have which points to the designated malware and Android device folder. In this fashion, teachers can have subsets of emulators and malware for specific classes and groups. To change what materials the student receives involves only changing where their symbolic link points to. This allows the environment to scale for larger deployments. Client machines obtain materials by using a created Bash script that utilizes the FTP client functionality of the wget utility to synchronize materials on the client with the server. This action, which can be performed manually by running the included script, is also done periodically through the use of the cron daemon in Linux. The cron utility runs the script every five minutes to ensure that materials are up-to-date. Along with synchronizing the Android emulators to the folder that the SDK is configured to read, scripts have also been created that download the emulators to a separate folder to serve as a fail-safe backup for students. These backups are protected using immutable bit within Linux, and scripts have been written that update the emulator backups from the server (nearly identical to the previously mentioned update scripts) as well as restore the backups to the android emulator folder that the SDK uses.

D. Configuration Tracking

Knowledge about the currently configured Android emulators on a client system could be beneficial to both students and professors in a lab environment. This could aid a professor in tracking a student's progress, and a student could be assisted with troubleshooting if the professor viewing the information spots a misconfiguration. To accomplish this, both client and server Python scripts have been created that transfer desired information from clients to the server. The configurations for android devices are stored in files within subdirectories of the directory created during installation of the software development kit. Each emulator has a main configuration file that is named based on the title given to the emulator during creation. A subfolder with a similar name contains other files necessary for the emulator, including some other configuration files. The client Python script pulls information from these files and sends this information to the server, which includes the following information: emulator name, android version, API version, and device type. This information can be expanded to include any other desired settings for lab operations. The server Python script receives information from the client script and writes this information to an HTML file in the appropriate format within the web directory of an Apache server. The HTML file is named based on the student name passed by the client script. Once the HTML files have been created and written to, the configurations can be viewing through a web browser on the server.

E. User Maintenance

All necessary users are implemented on the server virtual machine, which reduces management necessary for a lab environment. Currently, users have local accounts with UIDs higher than 1000, as the first 1000 are reserved for professors. To add a batch of users, a created script accepts a text file list of users separated by newline characters and creates the appropriate users on the system with default passwords and symbolic links as home directories. Once the duration of a course ends, it would be necessary to remove all these students in an easy manner. Another devised script removes all users and user-related data by eliminating all accounts with UIDs higher than 1000. This is appropriate for system configuration in an Ubuntu environment.

III. Future Work

While it might seem that this solution in its current state is not adequate for a lab environment, this is due to the fact that it is not a finished product. It is the authors' intend to continue working on this system until a solution can be provided that is desirable for any educational environment interested in Android malware research. Future work that will be addressed in the upcoming months include the following:

A. Configuration Reporting

To increase the usefulness of the configuration reporting feature, a few updates can be made to the current system. First, additional data can be passed from the client to the server to aid troubleshooting. Additionally, it would be beneficial to create a pseudo-versioning system for student configurations. This can be done by creating subdirectories within the web root on the server for each student, and then creating new HTML files for each day a student is active. This would allow a Professor to more effectively track a student's progress throughout the duration of a course. There should also be an option for a Professor to reset these logs for various reasons. It also might be necessary to configure authentication for this service to ensure the privacy of student data.

B. Administrative Tools

Although scripts may be adequate for a small deployment of this environment, this implementation does not scale well. To rectify this, a web administrative environment will be developed that provides an intuitive interface for management to configure their systems. It will include a means of setting multiple student symbolic links to a desired directory in an easy manner. This interface can call the previously written scripts to advance their utility. Any settings made through the management interface can be pushed to clients using SSH. Utility can also be increased by created a series of man pages, or at least a help parameter, for each script describing its purpose and usage.

C. User Management

Although the current means of managing users is adequate for an Ubuntu system, these methods might be incompatible with other Linux distributions. To mitigate this, user management scripts could be updated to detect the UID scheme used on the server system and adjust their operations accordingly. This way, there is no chance of making undesired changes to the users on the server system.

D. Emulator and Malware Synchronization

To accommodate different scenarios, it may be necessary to introduce a flag into the synchronization scripts that toggles if existing content on the client is removed before updating, or if the new content is simply appended to the existing content.

IV. Conclusion

Android-based malware is a hot topic that continues to assert itself as a primary concern within our cyber world. As the amount of Android devices continues to increase, the threat of malware targeting these devices will as well. A basic environment has been developed for an educational organization to introduce Android-based malware research in a lab format. This product is not a fully-fledged solution, but provides the groundwork for creating a comprehensive approach for facilitating research on Android-based malware.

V. Acknowledgement

The lead author would like to thank co-authors Professors Bill Stackpole and Tae Oh for their guidance throughout this study, and their continued support in creating this solution.

References

[1] Android SDK. Google, 2013. https://developer.android.com/sdk/.
[2] D. Goodin. Android's built-in malware scanner gets a failing grade. Arstechnica, December 2012. http://arstechnica.com/security.
[3] D. Kellogg. 40 Percent of U.S. Mobile Users Own Smartphones; 40 Percent are Android. nielsen.com, September 2011. http://www.nielsen.com.
[4] TrendMicro. TrendLabs 2012 Mobile Threat and Security Roundup: Repeating History, 2012. http://www.trendmicro.com.
[5] L. Whitney. Windows 8 swells to 2.7% of OS market. CNET, March 2013. http://news.cnet.com.

Keynote: Why Every CSO Needs to Know Industrial Control Systems (ICS)

Billy Rios
Director of Consulting, Cylance, Inc.
Chair of the Operational Security Testing Panel, NBISE

INDUSTRIAL control systems (ICS) have introduced tremendous cost savings by automating some of the enterprise's most critical operations. Do you understand the systems that support your critical data centers and corporate campuses? Do you understand the risks associated with these technologies? Every data center, large building, and corporate campus around the world plays host to environmental controls, building entry systems, safety systems, and many other automation systems that are considered ICS. In many industries these systems are a vital component to the enterprises most critical business operations. Given the complexity and specialization of these systems, many of these systems are managed and operated outside of the traditional IT sphere, leaving traditional vulnerability and risk management programs blind to their existence and the risk associated with these systems. Many of these systems are even managed and maintained by external third parties, providing a backdoor to your corporate network and hence represent a new weakest link in enterprise information security. Using the experience of a team with wide experience in critical infrastructure this session talks about strategies for understanding risk and implementing mitigating controls which need to be used to protect these vital systems.

Detecting Infection Source and Building Predictive Blacklists with an Attack-Source Scoring System

Liyun Li and Nasir Memon

***Abstract*—We present a network behavior based scoring system dedicated to inferring the maliciousness of hosts outside the perimeter of an institution/enterprise. Our viewpoint is strictly from the perspective of a network administrator. The scores are generated for external hosts outside the perimeter of the institution where the system is deployed, which we call "attack-source score". The unique property of our approach is that, we believe most infections stem from interactions with unknown external hosts with the assumption that the external hosts are "responsible" and "accusable" when its internal counterpart exhibits malicious/suspicious behavior. This unique feature of our approach makes our system independent of particular attack vectors and abstracts away attack characteristics. With a real deployment, we show by experiments that the system provides a global view of the maliciousness/risks of external hosts, and demonstrates the application of our system in two use cases: (1) detecting the infection source or a ranked list of suspected infection sources for network forensic and incidence response purposes; (2) building a predictive blacklist to aid network administrators to be aware of potentially dangerous external IPs even before an attack has been detected and the IP exhibited on third-party blacklists.**

***Index Terms*—Wireless University Serial Bus protocol, Security analysis, USB**

I. Introduction

With THE rapid growth and development of network technologies, detecting infected hosts within specific domains such as within an enterprise/institution does not necessarily imply being free of risk, since this does not preclude the existence of other infected hosts that are dormant since attacks have become more and more sophisticated and dynamic [1]. For example, when one internal hosts gets infected and behaves as a botnet zombie controlled by the botmaster, the actions of only quarantining and fixing this infected machine may not be sufficient since there might already be other hosts within the network infected and controlled by the same botmaster. As long as the infection source is still at large and accessible to internal hosts, further risk of infection still persists. Therefore, the effect of fixing existing infected hosts within an enterprise/institution is not sufficient without accurately pinning down and disabling access to the infection source by leveraging the malicious/suspicious symptoms exhibited by internal hosts.

Only with a "clean" view of the external world where most infectious sources are known, the perimeter of our network is then equipped with better security-awareness and thus becomes less vulnerable; attempts to interact with the external and potentially infectious hosts could be monitored more closely or blocked.

To detect infected and/or malicious external hosts, techniques such as reputation systems have been proposed. Most network reputation systems have used the concept of "network neighborhood [2], [3], [4], with the implicit assumption that nodes in the neighborhood of a known-malicious node could also be malicious, and have focused on defining neighborhoods and associated metrics that are appropriate for the chosen threat model. Inherently, the threats that can be detected by these proximity-based reputation systems are restricted to particular attack characteristics which predefine the proximity measure. Therefore the capability and applicability of these proximity-based reputation systems are limited by certain assumed attack vectors and such systems do detect any threats beyond the scope of predefined attack model or known maliciousness. Additionally, proximity-based reputation systems are inadequate to detect the infection source, since they mostly utilize similarities between the symptoms of the infected hosts and do not fully leverage the correlation of exhibited malicious behavior and historical interaction patterns.

An enterprise or institution could also utilize public blacklists [5], [6] or subscribe to private blacklisting services as an alternative to using reputation scores. However, an external malicious IP may not eventually show on any blacklist if the attack is targeted to specific networks. Even though the external malicious IP will be blacklisted by third-parties, there is a time delay in the sense that it gets blacklisted some time after it launches attacks to specific domains. Given these constraints of third-party public blacklists, it is desirable to have a customized global view of all the external unknown hosts, from the viewpoint of the protected network itself. In simple words, it is better to have customized predictive blacklisting rather than depending on third-party blacklists which could either not capture the malicious host or captures the malicious host with a time delay. Given predictive locally generated scores indicating the likelihood of an external host being the attack source and infectious, network administrators

L. Li is with Linkedin Corp Mountain View, CA 94043 USA (e-mail: liyli@linkedin.com)

N. Memon is with Polytechnic Institute of NYU, Brooklyn, NY 11201 (e-mail: memon@nyu.edu)

could determine the maliciousness of an external host even before any third-party blacklisting services captures it.

In this paper, we attempt to solve the problems of accurately detecting external attack sources and predicting potential sources of attacks by proposing a network behavior based scoring system designed strictly from an institution's perimeter point of view. Our system computes and generates *attack source scores* for all the entities outside the institutional boundary. More specifically, we want to leverage the "attack source" scores to solve two categories of problems: (1) Infection source detection: when our internal hosts exhibits malicious/suspicious behaviors, can we find out, if possible, the external infection source? Or perhaps a ranked list of the susopected external infection source? (2) Predictive Blacklisting: when an internal hosts tries to connect to an external host, how risky the external host is, as measured by our attack source scores. If the external IP is indeed malicious/infectious, can we detect and blacklist it before third-party blacklists?

To achieve these goals, we move away from the fixed concept of neighborhood, which is prone to false negatives (for highly distributed attack sources) and false positives (for nodes that happen to be in a "bad neighborhood"), and our attack-source scoring system on observed network traffic. The fundamental intuition is that, our internal hosts do get infected with a reason. And in most cases, infections stem from communicating with external hosts such as clicking on a phishing link or installing some malware unknowingly. Then intuitively, we charge "accusations" to all external hosts potentially responsible for an internal host's bad behavior, which is how we compute and propagate our attack source scores. Fundamentally, the attack source score "spreads" from known infected nodes to other nodes they have talked to.

A distinguishing feature of our system is the separation of information about a node's misbehavior from the information about the propagation of that misbehavior to other nodes. While most systems conflate the two by characterizing misbehavior of a known-malicious node and then detecting those same characteristics in other nodes, we use intrusion detectors and behavioral features to identify internal symptoms of maliciousness and then use distinct propagation rules based on observed traffic to derive the attack source score of that node's communication peers. This allows the creation of a system that is independent of particular attack vectors and that abstracts away attack characteristics to describe only the level of trust that can be afforded to each node in the network. We state our contributions as follows:

1) We propose an attack source score system from the internal network's viewpoint, where we leverage the fully-observed network behavior of our internal hosts to derive attack source scores for the external world

2) By quantifying the suspiciousness and maliciousness of internal host behaviors into a notion of "accusation score", we compute and propagate such "accusation scores" as attack source charges for all the external entities communicating with the internal hosts.

3) With two use cases, we experimentally show that the proposed attack source scores could be used in both discovering or ranking potential infection sources and predicting future malicious external hosts even before they get blacklisted by third-parties.

With our proposed system to infer the attack source score of any external host, thereby providing accurate charge for the most likely infection sources as well as risk management and control measures for network administrators, we hope that our methodology will bring alternative views and designs to the security community. In the rest of this paper, we describe our attack source detection score generation algorithm in detail and present the two experimental use case studies.

II. Related Work

Even though the generation of our attack-source score is very different from traditional reputational scores, they are still related and to some extent similar to reputation scores in the sense of measuring "maliciousness" of external hosts. In the area of network security research, reputation systems [7], [8] have been widely used to model the spread of attacks in a network, in particular because of their power to capture big-picture attacker activities and to provide a uniform view of the threat posed by network nodes [2], [4], [9], [10]. Many existing reputation systems are domain name based and built on various characteristics of DNS (prefixes and suffixes [3], WHOIS registration data [2], DNS server [4]) to determine whether a new name is "in the neighborhood" of a known-malicious name. The underlying logic is inherently proximity based: both malicious and benign hosts stay close to their own peers while keeping a distance away from peers in different category. Notos [2] is a dynamic reputation system based on domain name identities. It models malicious and benign domain names, and predicts a reputation score for any given domain name. The intuition is that benign and malicious DNS names, when mapped to different address spaces, have some significant differences. Exposure [3] also works entirely on passive DNS query data and focuses on the burst access nature of malicious domain name services from a temporal perspective. HPB [4] is an IP based blacklisting system. The system generates rankings of offensive IPs for each of the contributing institutes who share security logs. Many other techniques in botnet-detection could also been seen as assigning reputation scores to potential bot members [11], [12]. [8] provides a survey on the attacks and defenses of network reputation systems. Such scoring based reputation systems to measure risk and trust, are also used widely in area such as e-Commerce [13], [14], file-sharing systems [15], [16], and ad-hoc wireless networks [7]. [7] provides a survey of the various applications of a reputation system. Other work on detection infection/attack sources have mainly focused on other perspectives. For example, [17] studied the problem of tracing attack sources on the ISP level by leveraging router logs. In [18], the problem of preventing internal hosts from originating DDos attacks has been studied.

III. The Attack Source Scoring System for Tracing Maliciouness of External Hosts

Our attack source detection system works on network-flow data and is deployed at the boundary of an institution, thereby naturally separating all the hosts into two categories: the ones

inside our network perimeter are referred to internal hosts, while the ones outside our boundary are external hosts.

From an enterprise point of view, we fully observe the network activities of our internal hosts interacting with the external world, while for any individual external host we can only observe their traffic communicating with inside hosts. Therefore, we have a good picture of our internal hosts' behavior interacting with the external world. The idea of our approach is that: we believe that benign behavior is natural and inherent while there are reasons for malicious/suspicious activities. When one internal host begins to exhibit malicious/suspicious activities, the external hosts that this internal host has talked with should somehow be responsible and accused of a "attack source score" charge.

In other words, the behavior of the internal hosts is associated with their interactions with the external world. Hence an internal host is affected by the external host(s) he/she talked with and the external host(s) should be somehow responsible for our internal hosts behavior. More specifically, we compute attack source detection score based on such abnormal, suspicious or even malicious "behavioral" events triggered by network activities exhibited by our internal hosts. We quantify such behavior changes into additional attack source scores and propagate such scores through historical interactions between internal and external hosts. In addition, our system has the following properties:

1) Attack source scores of external hosts are from [-1,1], where negative is the malicious side and positive the benign side. Strangers start from a neutral score slightly towards the negative side (-0.1)

2) To bootstrap, IPs from whitelist [19] start from an initial attack source score of 0.9. IPs from blacklist [5], if shown in our traffic, starts from a score of -0.9.

3) Attack source scores gradually "decay" to neutral without evidence of observed traffic.

4) We memorize all the IPs that have hit a certain threshold of significant attack source score. And negative side attack source score gets forgotten much more slowly than the positive ones. If attack source score downgrade repeats, they get more compounded effect from previous attack source score.

5) Every time when internal host exhibits suspiciousness or maliciousness, the symptoms are quantified as "attack source accusation scores". And this charge score is subtracted from responsible external hosts to make their attack-source score closer to the negative side (i.e. more likely to be responsible as attack source).

A. Profiling Hosts Using Historical Behaviors

To accurately capture any suspicious behavior changes within our network perimeter for any internal host, we utilize the behavior features in Table-1 to monitor individual host's behavior. We compute the distributions of these features into four categories: workday-daytime, workday-offtime, weekend-daytime and weekend-offtime. These features are mostly based on network activities and could be directly computed from our network-flow for each internal host. All these features are computed within a time window *W*. To have enough resolution, the window length we use is 10 minutes. For example, we compute the unique number of IPs contacted by any internal host within the current time window. We also consider the ratios of successful TCP connection, as well as ratios of blacklisted IPs/Domains contacted by each internal host. With all these behavior feature values' distributions at different times, we profile each user with his/her own distributions. Any severe deviation from its behavior profile will be considered suspicious and under scrutiny. To abbreviate, we will refer to the vector of all these behavior feature values as $\vec{f}(T_k)$ in the rest of the paper.

TABLE I
SUMMARY OF SYMBOLS FOR BEHAVIOR FEATURES

Symbol	Meaning
nIP	Number of unique IP contacted
enPort	Entropy of distribution of opened ports
nTCP	Ratio of successful TCP connections / all TCP connections
nByt (r/s)	Number of bytes sent (received)
nPkt (t/s)	Number of bytes sent (received)
nAS	Number of AS systems contacted
nPeer	Number of unique peer (IP-Port) pair contacted
wBlkIP	Ratio of blacklisted IPs contacted/contacted IPs
wBlkDN	Ratio of blacklisted Domain Names contacted

B. Attack Source Accusations Scores Stemming from IDS Alerts and Suspicious Behavior Changes

Given each user's behavior profile, we assume that any large deviation from a user's behavior profile could potentially be malicious. We explicitly quantify such large deviations, along with any raised network IDS alerts [20], into a score, which we call "attack source accusation score". Formally

$$C(T_k, intIP_i) = \alpha DIST_{L2}(\vec{f}_{intIP_i}(T_k), \vec{f}(T_k)) + (1-\alpha)N_{T_K} \quad (1)$$

where the accusation charge at the time window has two contribution sources. One is from the deviation of the current behavior feature values from the historical behavior of the internal IP $intIP_i$, weighted by $\alpha = 0.5$. The other contributor is the normalized number of IDS alerts triggered by this in this time widow. Both of the $DIST_{L2}$ and NT_K are normalized into (0, 1). The function $DIST_{L2}$ computes the deviation of the current behavior compared with the internal hosts' profile, given whether the window is within the workday and/or daytime. To alleviate the effect of noise, the function $DIST_{L2}$ only produces non-zero accusation scores when the current time period feature vector $\vec{f}(T_k)$ is larger than the profile feature vector $\vec{f}$ by 3 times of the standard deviation. Note $\vec{f}$ must fall into one of the four predefined categories.

These accusations produced as described above are the source of raw negative charges placed on the attack-source scores of the external hosts. For each internal host, the accusations are allocated to the external hosts it has interacted with within a window W_{accuse}, as subtractions to the current attack source score. From the viewpoint of external hosts, each external host receives an "attack source accusation score" from every internal host it has been communicating with. This is how we transfer the suspicious behavior changes of our internal hosts into attack source accusation scores and then convert them to negative attack source scores ascribed to the responsible external hosts.

C. Attack Source Score Computation & Propagation

With the periodically generated attack source accusation scores from the internal hosts, every external host's attack-source-score is affected as we assume that it has to be, to certain extent, responsible for the abnormal/suspicious behavior changes of the internal hosts it has talked with. For example, when an internal host clicked on a malicious link and got infected as a p2p bot. Then suppose the infected host started to initiate many connections, for example to discover more of its botnet peers [12]. In this scenario, all the previous IPs that this internal machine has visited within some reasonable time window before the behavior change could be the infection source. Therefor we distribute the accusation scores equally to all these external hosts.

One may ask the question what if the infected host visited legitimate external entities, such as ***google.com,*** after infection. This issue is solved by the way we adjust the accusation scores. More specifically, the accusation score are adjusted by dividing a denominator which is the unique number of internal hosts visiting the external host. Since the benign ones tend to have more people talking to them who do not become infected and exhibit malicious behavior afterwards, external IPs for frequently visited benign nodes such as ***google.com*** will have a larger size of set Q_j^{Tk} than a malicious external host, where Q_j^{Tk} denotes the set of internal hosts communicating with external host j in time period T_k).

Compare two external websites ***google.com*** and ***somemaliciousness.com***. Suppose one internal host gets infected and an accusation C_{accuse} has been charged to both of these websites. However, it is highly likely that number of internal hosts which have visited ***google.com*** is much greater than the number of hosts which visited ***somemaliciousness.com***. In this way, the attack-source accusation score placed to ***google.com*** is alleviated by a larger denominator and thus ***google.com*** receives a much smaller accusation while the number of internal hosts visiting ***somemaliciousness.com*** will be much smaller and the accusation charge will be then more severe. Formally, the raw attack source score of external host j is computed as:

$$R_j^{T_k} = -\beta \frac{1}{\left|Q_j^{T_k}\right|} \sum_{intIP_k \in Q_j^{T_k}} C(intIP_k, T_k) + \exp^{-(1-\beta)} R_j^{T_{k-1}} \qquad (2)$$

The weight of β controls the weight between the previous attack-source-score and the accusation charged in the current time period. Note that without any explicit accusations from the internal hosts, the value of R_j will decay towards 0 (neutral) with the only contributor as its previous score. This is actually the most frequently occurring case as malicious behavior changes and triggered IDS alerts are not very frequent for most hosts.

However, given that some external IP's attack-source scores deteriorate to a significant extent, we record this IP in the database, and explicitly change its decaying factor β to a larger value, which will induce two effects. Firstly, with a large β value in Equation-2, new attack source accusation score will be more severe than previous, given same level of malicious behavior change. Secondly, the exponential decaying factor of its previous attack-source-score will decay much slower, with a smaller value of $1 - \beta$. In this way, if anything malicious has been recorded towards an external IP, new accusations will compound into more charges while its previous bad attack-source-score tends to maintain.

Given these raw attack-source-score computed for each external host, we can propagate these score values from the accused external nodes to other external hosts. However, we have to be careful with this propagation as our observed cross-boundary traffic only constitutes a small part of the external hosts' activities. Thus, simple propagating with respect to "network proximity" of external hosts such as IP-proximity might be biased because of our limited viewpoint of the traffic.

Instead, we compute the distance measure of two external hosts based on the similarity of their traffic patterns in communicating with our internals. Consider the graph $G = (V, E)$, where V represents all the external IPs and the weight w on edge $E(i, j)$ is computed in the following way: let I_i and I_j denote the set of internal hosts that external IP i and j have contacted during this time window. The weight w is the Jaccard measure of these two sets, i.e., $w = |I_i \cap I_i| / |I_i \cup I_{ij}|$. For each time period, the attack-source accusation scores are injected to the responsible external hosts and we propagate using the "dye-pumping" algorithm as described in [12].

IV. Experimental Use Cases

In this section, we present two use cases of our proposed approach: discovering the infection source and generating predictive blacklists for potentially malicious hosts.

A. Case 1: Finding Infection Source

To accurately find the infection source, we set up virtual machines in a subnet and mimicked routine user network activities by making the hosts randomly visit external-websites. In addition, we also make these hosts actually visit websites hosting malware. More specifically, the virtual machines were operated to download and install the Config-C from ***www.offensivecomputing.com***. After infection, we restrict any out-going connections from these infected machines to subnets within our network but allow their communication with the malware peers outside our network. In fact, within a few minutes after the virtual machines get infected, the infected hosts attempted to establish connections with around 40K hosts outside our network perimeter. This is captured as malicious behavior change by our system's internal behavior features and triggers the deterioration of the attack-source-scores of the external IPs that the virtual machines visited.

We monitored the attack-source-scores of the IPs for the external websites that these hosts visited. Fig-1 shows the attack-source-score of the infection source. To compare, we also traced the average attack-source-score of random hosts from either the entire external IP set(the blue line in Fig-1) and the external IPs visited by the infected internal hosts (the black line in Fig-1).

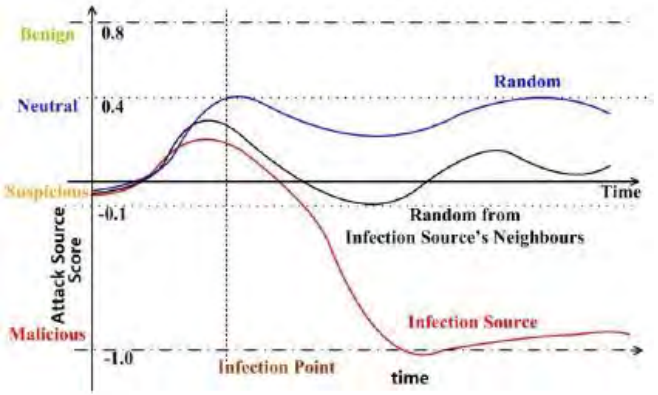


Fig. 1. Attack source score change when infection happens. The X-axis is the time and Y-axis is the attack source score.

As shown in the Fig-1, the attack-source-score of the infection source (***www.offensivecomputing.com***) is decreased much more significantly than the compared average random hosts. The point where the attack-source-score of the infection source gets decreased corresponds with the event where the malware was downloaded and installed. On the other side, average attack-source-score from 100 random hosts that the infected internal IP talks also gets worse. This is reasonable because when an internal host gets infected, all of its visited external hosts are suspicious for the infection to a certain extent. However, the attack-source-score of these hosts get better and approach the attack-source-score of random neutral non-malicious hosts (blue line) as time goes and no further malicious behavior gets exhibited. Therefore from this case study, we demonstrated via experiments that our proposed external-hosts based attack-source scoring system could discover correct infection sources.

B. Case 2: Predictivity of the Hosts with Worst Attack Source Scores

Another use case for the external attack-source-score system is to infer the unknown risks of the external world. Given the external hosts with worst generated attack-source-score (i.e. the most negative attack-source-score), we aim to verify whether they are indeed risky, or in other words potentially responsible for some of our internally observed malicious behaviors as well as any unobserved malicious behavior outside our network. Given access to limited traffic, one way to verify the attack-source-score is to check whether these IPs are also listed by Third-party blacklists such as Spamhause [5] or Barracuda [6]. Table II shows the top IPs with worst attack-source-score as we ran our system on one-week of network data and then traced them for two month. As shown in the table, the top worst attack-source-score IPs all have third-party reported malicious activities. The verification mainly comes from two categories. First we check these IPs against all the public-blacklists. In addition, we google these IPs on a daily basis and check the result. Some IPs are blacklisted by public blacklists, while others are complained and discussed associated with malicious activities from the Google-returned pages.

TABLE II
WORST ATTACK-SOURCE-SCORE IPS GENERATED BY OUR SYSTEM WITH IP, DOMAIN AND THIRD-PARTY VERIFICATIONS

IP	Domain Name	Blacklist
207.66.0.10	offensivecomputing.com	Infection Source
99.194.104.94	dyn.centurytel.net	Reported by: ThreatExpert.com
212.235.111.224	netvision.net.il	Subnet Dictionary Attack
89.178.231.5	corbina.ru	CASA, NOMOREFUNN
93.80.68.54	corbina.ru	BARRACUDA RATS-Dyna Spamhaus
94.75.193.168	bignaturalsonly.com	BARRACUDA RATS-NoPtr
95.79.194.228	UNKNOWN	Barracuda RATS-Dyna Spamhaus, Tiopan UCEPROTECT
94.179.142.100	pool.ukrtel.net	Barracuda RATS-Dyna Sorbs, Spamhaus, Tiopan
85.141.164.20	mtu-net.ru	Spamhaus
91.124.220.170	ukrtel.net	Reported by: BotsScount.com
83.22.174.108	adsl.tpnet.pl	Barracuda RATs-Dyna NOMOREFUNNSORBS CASA, Spamhaus

To quantitatively measure how many of the worst attack-source-score IPs are eventually blacklisted by entities besides us. We checked the top 10, 100 and 200 IPs with public blacklisting resources described above section. As shown in Fig 2, all the top-10 IPs are finally identified to be malicious in the corresponding blacklist from third-parties, while for the top-100 the ratio is around 95% and the top-200 is around 49%. Note that it is not necessary that all the top-200 external IPs we are necessarily malicious. However, the accuracy of our assigned attack source score lies in the fact that, if we make a ranking of the worst attack-source-score IPs between top-10 to 100, the precision that these external hosts have been infectious and exhibited offensive behaviors is more than 95% accurate. More importantly, our attack-source-score is based on a localized view and we assign significant attack-source-score to these IPs even before most of them get identified and blacklisted by other third-party services. Therefore, our generated attack-source-score is not only accurate but predictive in the sense that it will blacklist malicious hosts before one may find them on public blacklistings.

V. CONCLUSIONS AND FUTURE WORK

To conclude, we present a system for monitoring the maliciousness of hosts outside of an institution's perimeter. By leveraging the fully observed internal hosts' network activities, and assuming that most internal maliciousness has an external causal source, we compute attack-source-scores from internal hosts' behavior changes and accuse the external hosts they have talked to. The way we compute internal hosts' behaviors, along with assumptions that external hosts are responsible for internal hosts' malicious behaviors, makes the attack-source-score assigned by our system useful and accurate in two cases. One can not only use such scores for infection source

detection, but also build a predictive blacklist that records the potential malicious external hosts as soon as they communicate with internal hosts and potentially before they are captured by any third-party blacklisting services. Given the computation of attack-source-score, the infection source detection is limited to the fixed observation window within which all contacted external hosts are accused. While our window is long enough to capture typical infection sources, exhibited maliciousness after very long silence will make it hard to capture the accurate malicious external host and cause false-positives. To resolve this issue, dynamic and customized accusation window could be applied to each internal hosts depending on its traffic patterns. We leave this as future work.

REFERENCES

[1] K. Hoffman, D. Zage, and C. Nita-Rotaru, "A survey of attack and defense techniques for reputation systems," ACM Comput. Surv., vol.42, Dec. 2009.

[2] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster, "Building a dynamic reputation system for dns," in Usenix Security Symposium, 2010.

[3] L. Bilge, E. Kirda, C. Kruegel, M. Balduzzi, and S. Antipolis, "Exposure: Finding malicious domains using passive dns analysis," in Proceedings of NDSS, 2011.

[4] J. Zhang, P. Porras, and J.Ullrich,"Highly predictive blacklisting," in Usenix Security Symposium, ser. SS'08, 2008

[5] http://www.spamhause.org/zen/

[6] http://www.barracudanetworks.com

[7] M. Ibrohimovna and S. d. Groot, "Reputation-based systems within computer networks," in Proceedings of ICIW 2010, pp. 96-1-1.

[8] D. Z. K. Hoffman and C. Nita-Rotaru, "A survey of attacks on reputation systems," 2007

[9] S. Sinha, M. Bailey, and F. Jahanian, "Improving spam blacklisting through dynamic thresholding and speculative aggregation", 2010

[10] S. Hao, N. S. Ahmed, N. Feamster, A. G. Gray, and S. Krasser, "Detecting spammers with snare: spatio-temporal network-level automatic reputation engine," in Proceedings of NDSS,2009, pp.101-118

[11] G. Gu, R.Perdisci, J. Zhang, and W. Lee, "Botminer: clustering analysis of network traffic for protocol- and structure-level automatic reputation engine," in Usenix Security Symposium, 2008

[12] B. Coskun, S. Dietrich, and N. Memon, "Friends of an enemy: identifying local members of peer-to-peer botnets using mutual contacts," in Proceedings of the 26th Annual Computer Security Applications Conference, ser. ACSAC'10, 2010

[13] Gutowska and K. Buckley, "Computing reputation metric in multi-agent e-commerce reputation system," in Proceedings of ICDCSW 2008

[14] Jsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," 2006

[15] M. Siddharth, "A survey study on reputation-based trust management in p2p networks abstract," 2006

[16] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparision of peer-to-peer overlay network schemes," IEEE Communications Surveys and Turtorials, vol. 7, pp. 72-93, 2005

[17] T. Baba and S. Matsuda, "Tracing network attacks to their sources," IEEE Internet Computing, vol. 6, no. 2, pp. 20-26, Mar. 2002

[18] J. Mirkovic, G. Prier, and P.L. Reiher, "Attacking ddos at the source," in Proceedings of the 10th International Conference on Network Protocols, ser. ICNP'02, 2002

[19] http://www.spamhauswhitelist.com/en/

[20] http://www.snort.org

A Private Packet Filtering Language for Cyber Defense

Michael Oehler, Dhananjay S. Phatak and Alan T. Sherman

***Abstract*—Packet filtering is a central facet of cyber defense used to detect adversarial activity on a network. Detection stems from defensive efforts to discover new attack indicators, and efforts to share indicators with collaborating partners. There are instances where the sensitive nature of an indicator prohibits outright disclosure. Our private packet filtering language adapts the concept of private stream searching, and defines a new capability to filter packet data without revealing the indicator or result. The syntax of the language, the code to generate a private query, search and result, and the semantic constraints enforced by the language are presented. A cyber defender retains control of sensitive indicators, and coordinates a response action without revealing every indicator to the partner or risk disclosure to the adversary.**

***Index Terms*—Cyber Defense, Data Privacy, Oblivious Transfer, Packet Filtering, Private Search, Security Language**

I. Introduction

THE DISCOVERY of new cyber-attack indicators requires significant effort and expense. To ensure the greatest benefit, cyber defenders share new indicators with other collaborating partners (e.g., government and industry, corporations and their international subsidiaries.) However, indicators may be improperly disclosed by a partner, or exposed during an intrusion. This gives the adversary an opportunity to change their Tactics, Techniques, and Procedures (TTPs), reducing the value of the indicator. The defender is faced with a challenge. There is a need to share indicators and a requirement to control their dissemination.

Our contribution recognizes the association between this defensive challenge, and the capability provided by private stream searching. Specifically, we adapt the private search capability presented by Rafail Ostrovsky and William Skeith in 2005 [1], [2], and create a language for private packet filtering. Our high level language preserves the confidentiality of the indicator, and packets returned by the search.

Using our language, the defender constructs a query consisting of sensitive indicators, encrypts the query, and transfers the encrypted query (a filter) to the partner. The partner performs a private search on a stream of packets, and returns encrypted packets. If a matching packet is discovered, the defender notifies the partner of the adversarial activity, and coordinates a response. In this collaborative environment, the defender maintains situational awareness of adversarial tactics, controls which attack indicators are revealed, and advises the partner of current threat activity. This is a new scenario for cyber defense and private search.

The design of the language is intuitive and readable. For example, five lines define the cryptographic structure for a private search. Additional indicators and output buffers can also be specified in this structure. A single query can select different types of indicators and filter complex packet streams. This ability to search multiple indicators privately is unique.

Ostrovsky defined private stream searching. Bethencourt [3], [4] and Danezis [5] improved the storage efficiency of the output buffer. Yi constructed a conjunctive search [6], and Finiasz integrated Reed-Solomon codes with private searching [7]. We are also aware of Bethencourt's toolkit for private searching [8]. A high level language for *private stream searching* has not been previously formalized.

We name the language PPF for Private Packet Filtering.

II. Private Stream Search

In this section, we describe the salient features of private stream search. The terms, client, provider, document, and keywords are a generalization. Their use provides clarity, and permits an illustration of private search. In our context, these terms map to defender, partner, packets, and attack indicators, respectively. Last, the term, filter is retained, and a resulting collision in nomenclature is discussed at the end of this section.

A private search system preserves the confidentiality of the search criteria, and involves a client, and one or more information providers. A client generates a query, the provider performs the search, and delivers a response back to the client without gaining knowledge of the query or the result. The naïve approach transfers an entire data set from a provider to the client. Admittedly, this approach conceals the query from the provider. Ignoring bandwidth costs and a required client-side search, few providers would relinquish an entire data set (As an example, a partner is unlikely to divulge all network activity.) Alternatively, if the search criteria were kept secret, but knowledge of the result was evident, the structure of the query could be inferred. This is also unacceptable.

These concepts establish the fundamental properties of a private search system: the provider gains no knowledge of the query, cannot infer information about the query from the result, and client access is limited to results matched by the

Michael Oehler, Dhananjay Phatak, and Alan Sherman are with the Cyber Defense Lab, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, Maryland 21250 (email: {oehler1, phatak, sherman}@umbc.edu).

TABLE I
AN ILLUSTRATION OF PRIVATE STREAM SEARCHING

Client (Defender): Generates the query:		
Define a public dictionary:	$D = \{gelato, sherbet, snowball, sorbet, zebra\}$	See Fig. 1
Construct an encrypted filter:	$F = \{E(0), E(0), E(0), E(1), E(0)\}$	
Send D and F to the provider.		
Provider (Partners): Performs the search:		
Construct an encrypted buffer:	$B = \{\{E(0), E(0)\}, \{E(0), (0)\}\}$	See Fig. 2
Search using this document:	$d =$ "$unlike\ gelato\ sorbet\ has\ no\ calories$"	
Calculate a product of filter terms:	$s = \prod_{w_i \in d \in D} f_1 = f_1 \times f_4 = E(0) \times \mathrm{E}(1) = \mathrm{E}(1)$	
Calculate the search result as the exponentiation:	$r = s^d = E(1)^d = E(d)$	
Save the result to a buffer position:	$b_2 = \{s, r\} \times b_2$ $= \{\{E(1), E(d)\} \times \{E(0), (0)\}\}$ $= \{E(1), E(d)\}$	
Return the buffer to the client:	$B = \{\{E(0), E(0)\}, \{E(1), E(d)\}\}$	
Client (Defender): Processes the result:		
Decrypt the buffer:	$B = D(\{\{E(0), E(0)\}, \{E(1), E(d)\}\})$ $= \{\{0,0\}, \{1, d\}\}$	See Fig. 3
Recover the matching document:	$d =$ "$unlike\ gelato\ sorbet\ has\ no\ calories$"	

query [9].

Ostrovsky and Skeith created a clever private search system using (partial) homomorphic encryption [1], [2]. The system preserves the confidentiality of the search criteria, the result, and allows the client to match a document on a disjunction of keywords. The system is based on the asymmetric cryptosystem defined by Paillier, and utilizes the additive homomorphic property of the cryptosystem [10]. The cipher text from the Paillier cryptosystem is randomized. Thus, an encrypted value will be indistinguishable from another, even the same value, using the same public key. For instance, the encryption of $E(1)$ is indistinguishable from some other value of $E(1)$.

The client creates a list of encrypted ones and zeroes, corresponding to keywords of interest and non-relevant terms from a public dictionary; an encrypted filter. The client sends the filter and dictionary to the provider.

The provider performs the search by calculating a product of entries taken from the filter that associate with words in a document, an exponentiation, and a second product to save results to an encrypted output buffer. These calculations are performed on encrypted values (in the encrypted domain.) The provider is thus, unaware of the query or search-result. Furthermore, multiple documents may be stored in the buffer, creating a system that streams results, a private stream search (PSS) system.

Table I illustrates a simplified example. Consider a public dictionary D with five words, and a filter F containing five encrypted values. The fourth entry is an encrypted one $E(1)$ and expresses a private keyword that associates with "sorbet". The provider constructs the buffer B. The search entails a single document d. A product of filter entries f_1 and f_4, corresponding to words existing in the document and dictionary is calculated, and the exponentiation (A product of encrypted terms is equivalent to a summation of plaintext terms.) The provider then randomly selects a buffer position b_2. Results are saved to the buffer as a pairwise multiplication. The client decrypts the buffer, and recovers the document.

We transition to terms related to cyber defense (defender, packet, indicator), but will emphasize that the role and meaning of the encrypted filter is retained. In fact, an encrypted filter is integral to private search; "filter" appears as part of our syntax. A collision occurs when referring to a "packet filter." We define a packet filter as a graph of nodes (variables) used to select certain packets that are then sent to the private search system. For instance, a defender may not be interested in packets destined to common IP addresses. A packet filter could be constructed to disregard these packets, before sending the remaining stream of packets to the encrypted filter.

Our language parallels the three phases of a private search, and this is reflected in Table I with a reference to a code listing. The syntax and role of each listing is discussed next.

III. A PRIVATE PACKET FILTERING LANGUAGE

To introduce the central features of the language, we work through an example that searches for a sensitive indicator, a single Internet Protocol (IP) address.

Fig. 1 contains the code for query generation. This file is constructed by the defender, who defines the cryptographic

structures, indicators, and packet filter. The indicators stored in this code will remain private. The defender uses our parser to transform the query to a public form. This public code defines the private search, and is transferred to the partner. When all packets are searched, the partner transmits an output buffer of encrypted packets to the defender. The search is finished.

The creation of the query, the transformation, transfer, resulting packets, and any response action define the supporting process for an overall system. Our objective focuses on the definition of the language to support this process.

A. Query Generation

Fig. 1 shows the code for query generation. There are three portions: declarations, assignments, and an expression of a packet filter. For clarity, each portion is preceded by a comment, represented by the pound symbol.

```
# Declarations
key public paillier kPub {
    buffer outputBuf {
        filter in_addr dst malSite;
        filter in_addr dst c2IP
    }
};

graph myGraph {
    source file inFile;
    whitelist in_addr dst whList;
    whitelist in_addr src whtLst2;

    whitelist port dst whList3
};

# Assignments
kPub = { include "kPub.key" };
malSite = { 69.25.94.22  };
whList = {  192.168.0.0/16  };
whList2 = { 10.10.10.0/24,  11.11.11.11  };

# A packet filter
myGraph = {
    inFile -> whList -> whtLst2 :: malSite
};
```

Fig. 1. Query generation: sensitive indicators kept private.

Declarations: Variables are declared before use. Each is given a type, and may be followed by a qualifier. Furthermore, variable declarations are structured either to bind the cryptographic relationship of the filter variables, or to establish the function of a packet filter.

There are two declarations in Fig. 1: *key public paillier kPub* and *graph myGraph*. The first declaration expresses a public Paillier key, and is structured such that an output buffer and two encrypted filters are bound with the key. The qualifiers on the filters indicate that destination IP addresses will be searched. This key declaration establishes the cryptographic relationship used for a private search.

We note that the defender and partner use this public key to encrypted the filter and buffer respectively. The partner also uses the public key to perform the search. The defender uses the private key only to decrypt the buffer. This use of keys differs from that in a traditional public-key cryptosystem, which encrypts and decrypts a single document.

The second declaration establishes the packet filter. Specifically, we express a packet filter as a graph of nodes and edges. In this example, the graph variable is named $myGraph$, and includes the definition of four nodes, strictly four variables, one for data input, and three for data reduction. Data input is represented by the source declaration, and in this case input from a file is inferred: $source\ file\ inFile$. The remaining four declarations define whitelist nodes to discard packets of non-interest: $whList$, $whList2$, and $whList3$.

These node (variable) declarations will be bound with a filter variable via edge assignments. Together, all variables form a path, express a specific packet filter, and produce a private search system.

Assignments: The four assignments initialize the public key, a filter variable, and two whitelist variables. The public key is included from a file, and contains the modulus and public random integer, the public parameters of the Paillier cryptosystem. The filter variable, $malSite$ contains a sensitive indicator (A single IP address is used as a demonstration. Filter assignments typically contain a list of many indicators.) The remaining whitelist assignments depict two destination addresses, and a net block of source addresses to exclude from the private search.

Packet Filter: The final portion of Fig. 1 depicts the edge assignments of the packet filter. Variables are interconnected via the "–>" operator. We also introduce a sink operator, "::" to bridge nodes defined in the graph variable and that of the encrypted filter. The operator forms a path from the input source, $inFile$, passes packets through two whitelist variables, and then sinks packets in the filter variable, $malSite$ where the private search is performed. When executed, results are placed into the output buffer, $outputBuf$ that was previously related with the encrypted filter.

The network defender submits the code to the parser, transforms the indicators into a public form, and sends this public code to the partner.

B. The Search

The partner uses the public form of the code for the search as shown in Fig. 2. This listing displays a few alterations: The assignment of the filter variable, $malSite$, references an included file, $malSite.fltr$. This file contains the encrypted values of the indicators, and is also publicly releasable. A default assignment for the input variable, $inFile$, will prompt the partner for the name of a packet capture (PCAP) file.

Two additional lines define processing parameters for the output buffer. These parameters are produced by the parser in

lieu of a buffer assignment. The parameters specify the size of the buffer and that the partner's system constructs the buffer locally (i.e., The partner's system will encrypt a list of 1024 zeroes using the public key associated with the buffer.)

```
# Declarations
key public paillier kPub {
    buffer outputBuf {
        filter in_addr src malSite
    }
};
graph myGraph {
    source file inFile;
    whitelist in_addr dst whList;
    whitelist in_addr src whList2
};

# Assignments
kPub = {
    include "kPub.key"
};
malSite = {
    include "malSite.fltr"
};
inFile = {
    "Enter a PCAP Filename: "
};

whList = 192.168.0.0/16;
whList2 = { 10.10.10.0/24, 11.11.11.11 };

outputBuf.bufferSize = 2048;
outputBuf.production = local;

# Graph Execution
myGraph = {
    inFile -> whList -> whList2  ::  malSite
};
```

Fig. 2. The Search: public code for private packet filtering.

C. The Result

After completing the search, the partner's system will create an output file consisting of a PCAP header, represented by $0xa1b2c3d4$ ... and a buffer. The partner sends this file to the defender, who decrypts the buffer and assembles the matching packets into a PCAP file. The defender can then use additional packet processing tools. For brevity, only one ASCII hex value from the buffer is displayed in Fig. 3:

```
# The output buffer: A PCAP header
# and an encrypted buffer
outputBuf={
    0xa1b2c3d4…,
    {
       0x112233445566778899aa…
    }
};
```

Fig. 3. The Result: an encrypted output buffer.

Our application of private search exercises a special case. When a packet is searched, a single indicator (the destination address in this example) from a packet is tested against the encrypted filter. No more than one indicator can match for any given packet. Whereas in the general case, multiple words from a document could match, which scales a document by a constant factor. In our case, this scaling factor will be one, and does not have to be transferred to the defender. Our output buffer can thus, be initialized as a simple list of encrypted zeroes, and returned as a list of encrypted results without scaling.

IV. Syntax and Semantics

This section presents the formal syntax, private comments, processing parameters, and describes two principal and semantic tests: the variable and packet filter check.

Definition of the Syntax: The Appendix presents the syntax of the language in a traditional Backus-Naur Form (BNF): nonterminals are represented in a braced form, "<nonterminal>", and productions are presented with a single nonterminal on the left side of the "composed of" operator, "::=". Reserved words are in boldface. The syntax deviates slightly from tradition with the utilization of a regular expression range operator, and a repetition operator, for example "[a-z]" and "[0-9]{1,3}" to represent characters from the alphabet, and one to three digits, respectively.

Optional items (qualifiers) are bounded by square brackets. Curly braces "{" and "}", are terminals used to delimit the declaration of a key, buffer, or graph variable, and additionally when a list of values is required. We also use a state designator to bind the context of an assignment. For instance, the value assignment for an IP address variable is restricted by the "$ipInputState" designator. Finally, the start symbol, $< ppfProgram >$ defines our language as statements of declarations, assignments, and comments.

Private Comments: Within the syntax, there is a notion of public and private comments. When manipulating sensitive indicators, defenders are likely to attribute activity by intelligence source, origin, and by other characteristics of a named intrusion set. These comments need to remain private, are designated by a double pound, "##", and removed by our parser. Public comments, those that can still add clarification and can be sent to the partner, are defined by a single pound. For instance, the private comment in Fig. 4 identifies the indicators and attributes these indicators to a named threat actor. They are removed. The public comment remains in the

parsed version of the code. As a semantic rule, public comments are associated with the next variable, and in their order of appearance.

```
## Malicious hosts attributed to named
##threat actor, Fuzzy Bunny
##
# A filter assignment
malSite.obfuscate=true;
malSite={ 69.25.94.22 };
```

Fig. 3. Public and private comments.

Processing Parameters: Processing parameters define additional options.

The code fragment in Fig. 4 also depicts a processing parameter: the $malSite.obfuscate$ statement indicates that the variable name will be obfuscated in the public code (an obfuscated variable name is expressed as the Base64 string of a cryptographic hash.) This facet reduces a burden on the defender. Variable names can conform to practice, readability, and operational context in the private form, and presented in a non-revealing manner in the public form of the code.

In this instance, the obfuscation was localized to a specific variable. If the variable name had been excluded, the processing parameter is applied globally. For instance, $.obfuscate = true$, obfuscates the names of all variables.

The design includes three parameters that can be used by any type variable, $.obfuscate$, $.cwd$, and $.dataMap$, four parameters for buffer variables, and one for filter variables. The parameters are shown in Fig. 5.

```
# Applies to all variables
. cwd = <aDirectoryPath>
. datamap = [ include | inline ]
. obfuscate = [ true | false]

# Applies to buffer variables
. bufferS ize = <integer >
. production = [ remote | local]
. reuse = [ true | false]
. trigger = <integer >

# Applies to filter variables
. expand = <integer >
```

Fig. 4. Processing parameters for variables

The $.cwd$ parameter sets the working directory for output, and $.datamap$ indicates whether a variable's assigned data will be saved to an include file or inline with the code.

The $.bufferSize$ parameter establishes the number of entries in an output buffer. The $.production$ parameter indicates where the buffer is produced, either locally on the defender's system, or remotely on the partner's system. The $.reuse$ parameter specifies whether an initial copy of the buffer can be used again, after the maximum number of buffer insertions is reached. This threshold is expressed as a multiplicative factor of the buffer size, and specified by the $.trigger$ parameter. The $.expand$ parameter scales the filter, to reduce false positives. These parameters are discussed further in Section V, Facets of Design.

Variable Assignment Checks: After declaration, variables are assigned and then used in a packet filter, via edge assignments. These two states assigned and on a path, are tracked as part of the transformation to the public form of the code. A warning or error is generated if either one of these two states is not met, and as shown in Table II.

TABLE II
VARIABLE ASSIGNMENT CHECKS

Variable	Assigned?	On Path?	Result
filter	No	No	Warning Unused
	No	Yes	Error. Halt
	Yes	No	Warning Unused
	Yes	Yes	OK. Encrypted Filter
node	No	No	Warning Unused
	No	Yes	Warning No-Op
	Yes	No	Warning Unused
	Yes	Yes	OK. Use in Packet Filter

If a variable is not on a path, (the variable is not used in a packet filter), the variable is deemed unused, even if data has been assigned to the variable. A warning will be issued, indicating that the variable must be part of a packet filter, and then removed from the public code. This exclusion has no effect on the private form of the code, minimizes structure in the public code, and does not alter the intent of the search. This was the case for the whitelist variable, $whList3$ and the filter, $c2IP$ from Fig. 1. Neither variable appears in Fig. 2. A greater challenge occurs when a variable is left unassigned, but used in the graph. This condition results in an error or warning, depending on the type of variable.

A design decision was made to produce an error when an unassigned filter variable is used in a graph. The error halts processing. While it is possible to construct an encrypted filter without search criteria, the search would consume resources without producing a result. This seemed inefficient, but in a strict sense, this decision precludes the ability to perform a null or empty search.

When a node variable is left unassigned, but utilized in a graph, a warning is issued. However, the variable, its declaration, and use in the graph remain. Packets pass through this unassigned node, when the packet filter is executed, and without modification. This is a design decision to support a no-operation (no-op). As packets traverse the packet filter, the no-op imparts little impact.

Packet Filter Validation: The semantic check for graph correctness assures that a path starts at a source variable and sinks to a filter variable. The in-degree of each node must be one, excluding source nodes which have an in-degree of zero. A filter variable cannot connect to another variable. This prohibits feedback loops in the packet filter. These rules are depicted in Figure 1 with three packet filters and their visual representations for clarification. The variables are from Fig. 1. The packet filter in Fig. 6-a for $myGraph$ is correct. The edge assignments start at a source variable, $inFile$, and sink to the

filter, $malSite$. The in-degree of each node is one. This is a valid packet filter. Fig. 6-b also has a correct syntax, specifies a source and sink, and has valid edge assignments. A warning though will be issued, since whList3 was not assigned any data. An error is additionally issued and processing halted since the filter variable, $c2IP$ was not assigned; notice that no indicators for this variable appear in Fig. 1. Fig. 6-c demonstrates an invalid path because the in-degree to the filter variable is two. Processing is halted.

Last, we note that our graph representation is similar to the edge operator (edgeop) for directed graphs in the GraphViz language [11], [12], and the ability to select, direct, and reduce traffic volume is a common data processing paradigm. For example, the rwfilter command selects specific Netflow records in the SiLK tool suite, and the rwsender command creates a tee, directing data to multiple receivers [13], [14]. Our graph processing approach also shares functional similarities with Unix pipes.

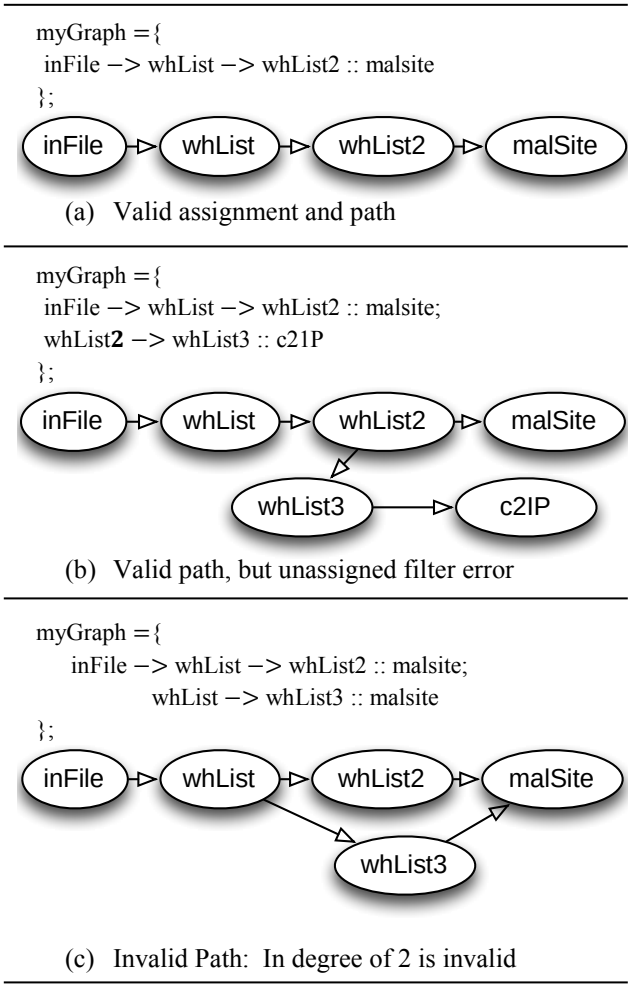


Fig. 6. Three representations of a packet filter.

V. Facets of Design

There are some design decisions that are reflected and apparent in the syntax. In this section, we will address the less apparent decisions, including how filter entries are referenced, a filter size is selected, and how a buffer is managed.

Encrypted Filter References: In the original private search system, entries in an encrypted filter were associated with a public dictionary of words [1]. Consider the contrived public dictionary D from Table I.

This is an example of the inference problem, forming conclusions from premises without authorization [15].

While the associated entries in the filter F are encrypted, precluding exposed interest in the keyword "sorbet", the overall interest in frozen desserts is evident. For this reason, a dictionary is assumed to be diverse, if not unabridged. The solution works for common nouns, general terminology, etc. However, proper nouns and domain specific terms are not as easily obfuscated. Exposure in a small set may be sufficient to divulge knowledge, and enumerating the full set may not be possible.

Indicators are domain specific and cannot be exposed in a public dictionary, even if the indicators were intermingled with a large number of unrelated (chaff) indicators. The adversary need only look for their address, domain name, etc. Furthermore, it may not be possible to enumerate every indicator. Our design does not reference filter entries through an association in a public dictionary.

Bethencourt detailed a method to eliminate the dictionary [4]. The value of a truncated cryptographic hash is utilized as an index into the filter, and the one-way property of the hash assures that the keyword cannot be inferred. A cryptographic hash will also exhibit a uniform response for all inputs, assuring that index values are generated uniformly. Our design utilizes this approach when referencing filter entries.

There is a drawback. That is, for a given hash function $H(x)$ and two words, w and w', hash values may collide $H(w) = H(w')$. This is not (generally) an issue for full-length cryptographic hash values, but the reduction of the hash space will introduce false positives.

Filter Expansion: To counter false positives, the filter size must be increased to an acceptable size. We use an expansion factor relative to the number of indicators. Unfortunately, this leads to a quadratic relationship in the size of the filter. A defender analyzing a thousand IP indicators across multiple intrusion sets, and in an environment which requires little or no spurious access to data, for instance at a rate of one in a thousand, produces a filter of a million entries.

Filter expansion is reflected in the design as the $.expand$ processing parameter for filter variables (Fig. 5). The parameter indicates the proportional expansion of indicators to determine the size of the filter. The parameter is not presented in the public form of the code so that the number of sensitive indicators cannot be immediately deduced.

Buffer Management: Some packets will match on an encrypted indicator in a filter, and will be returned as an encrypted result in the output buffer. Other packets will not match; the result of this non-matching search is an encrypted zero. Informally, an output buffer is not changed when a zero, in the plain text domain, is added to the buffer. However, the partner is unable to detect whether a packet matched and is stored to the buffer, or not. Since the result is encrypted, this leads to a conundrum. The partner does not know what was

stored, which buffer positions are available, or when to stop.

Current storage strategies thus, employ a randomized approach that is fundamentally based on the color survival theorem. Ostrovsky gives a formal presentation of this theorem [1], but intuitively, the strategy saves the result to a few randomly selected buffer positions. The occurrence of multiple copies will perpetuate the survival of at least one copy. A non-match has no effect on the buffer. In Ostrovsky's approach, the client recovers documents from the buffer by decrypting, and then searching for surviving documents. However, at the buffer's limit, some positions may be chosen multiple times, eliminating surviving copies. The recovery algorithm has a non-zero probability that all copies will be overwritten. Large buffers may minimize this condition, but this results in storage inefficiencies.

Subsequent research has sought to improve the document storage algorithm and therefore, document recovery rates. The research recognizes that a collision in a buffer position results in a linear combination of documents, and as a direct result of the homomorphic property of the Paillier cryptosystem. Information is not entirely destroyed, only obscured. Research has thus, qualified external structures, additional processes, and leveraged the redundancy of multiple copies to extract documents that were not recoverable in the original approach.

Bethencourt deconstructs the linear combination through a series of linear equations, but requires a second encrypted buffer [4]. The second buffer acts as a Bloom filter, when decrypted, and validates a document's membership in the output buffer. This knowledge can then be used to establish a system of linear equations to solve.

Danezis presents a (simple) iterative method: identify the singletons, calculate the positions that those documents were stored too, subtract the document value from those buffer positions, and repeat until no further singletons are discovered or the buffer is empty [5]. The use of the term, singleton was defined by Finiasz for this context [7]. The approach does require a function that duplicates document positions by the defender and partner. A (truncated) hash of incremented document values was suggested: $positions = \{H(d_i), H(d_i + 1), H(d_i + 2), \ldots, h(d_i + l)\}$, for each document d_i and for a pre-determined number of copies l. This algorithm replaces Ostrovsky's randomized approach for buffer position selection. Danezis's iterative method achieves full recovery when three document copies are utilized, and the total number of matching-documents inserted into the buffer does not exceed half the buffer size, $m = 0.5\ x\ |B|$.

As a design decision, our prototype utilizes Danezis's iterative method to recover documents from the buffer. The simplicity and acceptable recovery rates justified use. However, this still does not address when the provider should stop inserting results into a buffer. If we abide by the theoretical results, a buffer, twice the size, is returned after every "m" insertions. This is unacceptable when the majority of packets never match.

We resolve this issue with a processing parameter, $.trigger$. The parameter specifies the number of insertions as a multiplicative factor of the buffer size, for example 100, 1000, or 10000 insertions occurs before returning the buffer.

VI. Experimentation

We implemented a working model of the language. Our prototype consists of a lexical analyzer (Flex), a Bison parser, C++ code, and used *Mathematica* for the private search operations.

The result is three programs, ppf-generate, ppf-search, and ppf-recover that derive from a single code base to generate the query, perform the search, and retrieve results.

One challenge remained. We needed an experimental dataset, and one that does not impinge on operational data. We acknowledge the pursuit of a standardize corpora for security research [16]. While Garfinkel's focus is on digital forensics, his scenarios include network datasets, including the "Nitroba University Harassment Scenario" [17]. This data set contains 91,144 IP packets.

Fig. 6 shows the execution of the private search detailed in this paper. The listing depicts a collaborative environment consisting of a defender and partner system, and shows the sequence of commands executed on each system. The search determines if any traffic in the Nitroba data set is destined to 69.25.94.22. A fictional organization with known malicious intent operates a web server, $www.willselfdestruct.com$, at this address. This fictional IP address is a sensitive indicator, and is not initially revealed to the partner.

The dataset consists of inbound and outbound traffic to fifteen private netblocks. Since the intent is to reveal malicious outbound traffic, the query from Fig. 1 includes a whitelist to ignore any inbound traffic; packets sent to 192.168.0.0/16 are dropped. In total, our prototype processed 45,776 IP packets.

The result of this search revealed 101 packets destined to 69.25.94.22. The defender gains situational awareness, and initiates a response action. If deemed appropriate, the defender may reveal the IP indicator and activity to the partner. We emphasize that all other indicators remain private.

Computation is bound by the number of modular exponentiations performed during the search. As depicted in Table III, query generation requires an encryption for each filter entry, and a result is obtained after decrypting the output buffer. The computational cost to create an encrypted filter is $O(|F|)$ encryptions, and the output is recovered in $O(|B|)$ decryptions.

Recall that each exponentiation (a result) is saved to the buffer three times (as per the color survival theorem.) Hiding this constant, the cost to save results to the buffer is $O(|T|)$ modular multiplications. Last, the computational cost for buffer construction can be performed off-line, and is not an overall factor.

The search however, must partition each packet to a set of values with a bit length less than the length of the modulus. The number of exponentiations is thus, hidden by another constant factor, but in general $B \approx F << T$ where T is a data set of packets. The computational cost of $O(|T|)$

exponentiations outweighs that of other operations in this system.

TABLE III
COMPUTATIONAL COST

Query Generation	
Filter Construction	$O(\|F\|)$ Paillier encryptions
The Search	
Buffer Construction	$O(\|B\|)$ Paillier encryptions
Packets Searched	$O(\|T\|)$ Modular exponentiations
Saving the Result	$O(\|T\|)$ Modular multiplications
The Result	
Buffer Decryption	$O(\|B\|)$ Paillier decryptions

VII. CLOSING REMARKS

We developed a high level language for private packet filtering (PPF). The language adapts the concepts of private stream searching, preserves the confidentiality of sensitive indicators, and is highly suited for cyber defense in a collaborative environment. Using our language, a cyber defender maintains situational awareness, controls which indicators are revealed, and shares threat details with collaborating partners.

We designed the language to be user friendly and to bridge the cryptographic constructs of private searching in a form applicable for a cyber defender. This paper presents the syntax of the language and demonstrates a private search for a sensitive IP address.

A greater breadth of searchable indicators is also possible. Additional filter types, such as $filter\ smtp\ subject$, $filter\ http\ user-agent$, $filter\ dns\ a-record$, etc. can be constructed as future work.

The language can also be adapted for new file scanners, anti-virus, and other defensive products that search for malicious content without revealing knowledge of the search.

```
defender$ ppf-generate -r privateIndicators.ppf -w public.ppf
defender$ echo "Send public.ppf to the partners."

partner$ ppf-search -r public.ppf -w buffer.ppf
  Enter PCAP filename:    nitroba.pcap
partner$ echo "Return the buffer file, buffer.ppf"

defender$ ppf-recover -k kPrivate.key -r buffer.ppf -w partnerActivity.pcap
defender$ tcpdump -n -c 2 -r partnerActivity.pcap
01:03:43.729507 IP 192.168.15.4.35984 > 69.25.94.22.80: Flags [S], seq 3033670331, win 64240, options [mss 1460 …
01:03:43.819342 IP 192.168.15.4.35984 > 69.25.94.22.80: Flags [.], ack 2749676331, win 64296, options [nop,nop,TS val …
01:03:43.825871 IP 192.168.15.4.35984 > 69.25.94.22.80: Flags [P.], seq 0:526, ack 1, %win 64296, options [nop,nop,TS val …

The remaining packets from the tcpdump are not shown for brevity.
```

Fig. 5. A Demonstration of the Private Packet Filtering Prototype

APPENDIX A

The BNF Representation of the Private Packet Filtering (PPF) Language:

```
<ppfProgram>            ::=  <statements>
<statements>            ::=  <declaration> | <assignment> | <comment>

<declaration>           ::=  <keyDeclaration> | <graphDeclaration>
<keyDeclaration>        ::=  key public paillier <variable> { <bufferDeclarations> };
<bufferDeclarations>    ::=  <buffer> | <buffer>; <bufferDeclarations>
<buffer>                ::=  buffer <variable> { <filterDeclarations> };
<filterDeclarations>    ::=  <filter> | <filter>; <filterDeclarations>
<filter>                ::=  filter in_addr [src|dst] <variable>
                               | filter port [src|dst] <variable>

<graphDeclaration>      ::=  graph <variable> {<nodeDeclarations>} ;
<nodeDeclarations>      ::=  <node> | <node>; <nodeDeclarations>
<node>                  ::=  source [file|interface] <variable>
                               | whitelist ip [src|dst] <variable>
                               | whitelist port [src|dst] <variable>

<assignment>            ::=  <varAssignment> | <parameterAssignment>
<varAssignment>         ::=  <variable> = <value> | {<values> | include "<fileName>"};
<parameterAssignment>   ::=  <variable>.<parameter> = <pValue>;   | .<parameter> = <pValue>;
<fileName>              ::=  <text>
<pValue>                ::=  <decimalValue>
<values>                ::=  <value> | <value>, <values>
$numInputState<value>   ::=  0x<hexValues> | <decimalValues>
$edgeInputState<value>  ::=  <variable> ->  | <variable> ::   | <variable>
$ipInputState<value>    ::=  <ipAddresses>
<ipAddresses>           ::=  <ipAddress> | <ipAddress>, <ipAddresses>

<comment>               ::=  <publicComment> | <privateComment>
<publicComment>         ::=  #<text>
<privateComment>        ::=  ##<text>

<parameter>             ::=  cwd | datamap | obfuscate | bufferSize | production | reuse | trigger | expand

<variable>              ::=  <variableID>
<variableID>            ::=  <letter> | <variableID><letter> | <variableID><digit>
<ipAddress>             ::=  <netBlock> | <dottedDecimal>
<netBlock>              ::=  <ddigit>{1, 3}.<ddigit>{1, 3}.<ddigit>{1, 3}.0/24
<dottedDecimal>         ::=  <ddigit>{1, 3}.<ddigit>{1, 3}.<ddigit>{1, 3}.<ddigit>{1, 3}
<decimalValues>         ::=  <decimalV alue> | <decimalValue><decimalValues>
<decimalValue>          ::=  <ddigit>
<hexValues>             ::=  <hexValue> | <hexValue>, <hexValues>
<hexValue>              ::=  <hdigit>
<text>                  ::=  <character> | <character><text>
<character>             ::=  <letter> | <ddigit>
<letter>                ::=  [a−z] | [A−Z]
<hdigit>                ::=  [0−9] | [a−f]
<ddigit>                ::=  [0 − 9]
```

REFERENCES

[1] R. Ostrovsky and W. Skeith, "Private searching on streaming data," in Annual International Cryptology Conference (CRYPTO'05), vol. 3621, pp. 223–240, 2005.

[2] R. Ostrovsky and W. Skeith, "Private searching on streaming data," Journal of Cryptology, vol. 20, no. 4, pp. 397–430, 2007.

[3] J. Bethencourt, D. Song, and B. Waters, "New construction and practical applications for private stream searching (extended abstract)," in IEEE Symposium on Security and Privacy (SP'06), pp. 132–139, 2006.

[4] J. Bethencourt, D. Song, and B. Waters, "New techniques for private stream searching," ACM Transactions on Information and System Security (TISSEC), vol. 12, no. 3, 2009.

[5] G. Danezis and C. Diaz, "Space-efficient private search with applications to rateless codes," in Financial cryptography (FC'07), pp. 148–162, 2007.

[6] X. Yi and E. Bertino, "Private searching for single and conjunctive keywords on streaming data," in 10th annual ACM workshop on Privacy in the electronic society (WPES '11), pp. 153–158, 2011.

[7] M. Finiasz and K. Ramchandran, "Private stream search at the same communication cost as a regular search: Role of LDPC codes," in Proceedings of the 2012 IEEE International Symposium on Information Theory, pp. 2566–2570, IEEE, 2012.

[8] J. Bethencourt and B. Waters, "Private stream searching toolkit." http://acsc.cs.utexas.edu/, 2011.

[9] S. M. Bellovin and W. R. Cheswick, "Privacy-enhanced searches using encrypted bloom filters," Tech. Rep. CUCS-034-07, 2007.

[10] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'99), vol. 1592, pp. 223–238, 1999.

[11] E. Gansner, E. Koutsofios, S. C. North, and K.-P. Vo, "A technique for drawing directed graphs," IEEE Transaction Software Engineering, vol. 19, no. 3, pp. 214–230, 1993.

[12] E. Gansner, E. Koutsofios, and S. North, "Drawing graphs with dot, the dot user manual," tech. rep., 2006.

[13] C. Gates, M. Collins, M. Duggan, A. Kompanek, and M. Thomas, "More netflow tools: For performance and security," in Large Installation System Administration Conference (LISA '04), the Eighteenth Systems Administration Conference, 2004.

[14] T. Shimeall, S. Faber, M. DeShon, and A. Kompanek, "Using SiLK for network traffic analysis," tech. rep., CERT Network Situational Awareness Group, 2010.

[15] M. Collins, W. Ford, J. O'Keefe, and B. Thuraisingham, "The inference problem in multilevel secure database management systems," in 3rd RADC Database Security Workshop, The MITRE Corporation, 1990.

[16] S. Garfinkel, P. Farrel, V. Roussev, and G. Dinolt, "Bringing science to digital forensics with standardized forensic corpora," Digital Investigation, vol. 6, pp. S2–S11, 2009.

[17] S. Garfinkel, "Digital corpora producing the digital body – nitroba university harassment scenario." NSF DUE-0919593, http://digitalcorpora.org/, 2011.

Cloud Security: Attacks and Current Defenses

Gehana Booth, Andrew Soknacki, and Anil Somayaji

***Abstract*—This paper presents a high-level classification of current research in cloud computing security. Unlike past work, this classification is organized around attack strategies and corresponding defenses. Specifically, we outline several threat models for cloud computing systems, discuss specific attack mechanisms, and classify proposed defenses by how they address these models and counter these mechanisms. This examination highlights that, while there has been considerable research to date, there are still major threats to cloud computing systems, such as potential infrastructure compromise, that need to be better addressed.**

***Index Terms*—Cloud Computing, Security, Virtual Machines**

I. Introduction

CLOUD COMPUTING is now the foundation of most Internet usage. Email, search engines, social networks, streaming media, and other services are now hosted in "the cloud"—large collections of commodity servers running coordinating software that makes individual hosts largely disposable. While cloud computing has lowered costs and increased convenience, the accessibility and centralization of cloud computing also creates new opportunities for security breaches.

Many security researchers have studied various aspects of cloud computing security from both an offensive and defensive perspective. In this paper we give a high-level classification of this work in order to examine to what degree proposed defenses can address different kinds of cloud-specific attacks. Specifically, we organize the cloud security literature into five areas: colocation denial of service, colocation breaches of confidentiality, data integrity and availability, data confidentiality, and infrastructure compromise. As we will show, while there has been significant progress, there remain major shortcomings in cloud defenses, even from the perspective of published research. While there have been other cloud security surveys and classifications published, ours is the first one organized around cloud-specific attacks and defenses. Our hope is that this survey can help guide researchers to work on areas of cloud security that have been less studied.

The rest of this paper proceeds as follows. Section II gives a bit of background on cloud computing. Section III describes our assumptions about the nature of threats against cloud computing as opposed to other computing platforms. Then in Sections IV, V, VI, VII, and VIII, we survey published attacks and defenses regarding our five attack areas–colocation denial of service, colocation breach of confidentiality, data availability and integrity, data confidentiality, and infrastructure compromises. Section IX details related work to our survey. Section X discusses the limitations we found in the literature and potential areas for future research.

This work was supported by Canada's Natural Sciences and Engineering Research Council (NSERC) through the ISSNet Strategic Network and Discovery Grants programs.

The authors are with the School of Computer Science, Carleton University, in Ottawa, Canada. (contact email: soma@scs.carleton.ca)

II. Cloud Computing

As outlined by Mel and Grance [1], cloud computing generally has five characteristics:

1. **Resource pooling**

   The provider's resources are pooled and shared between multiple customers.

2. **Broad network access**

   These resources are accessible through standard network protocols over the Internet.

3. **Rapid Elasticity**

   In a matter of minutes resources may be provisioned to scale out and released to scale in.

4. **Measured service**

   The provider measures and generally charges for usage of CPU, memory, disk, network bandwidth, or other resources.

5. **On-demand self-service**

   Resources can be provisioned via automated mechanisms

While every type of cloud service has these characteristics at its core, the various service models differ drastically in both form and function. We focus on three main service models: infrastructure as a service, software as a service, and platform as a service. Infrastructure as a service (IaaS) is the most basic service model for delivering cloud capabilities. Typically the consumer is given access to processing, storage, networks, and other resources necessary to run and/or deploy arbitrary software in a form that is close to having on-demand access to an arbitrary number of network-connected servers. An arbitrary number of "virtual servers" are multiplexed onto the providers' fixed number of physical hosts, generally using virtual machines (VMs) running on hypervisors. An example of IaaS is Amazon's Elastic Compute Cloud (EC2) service:

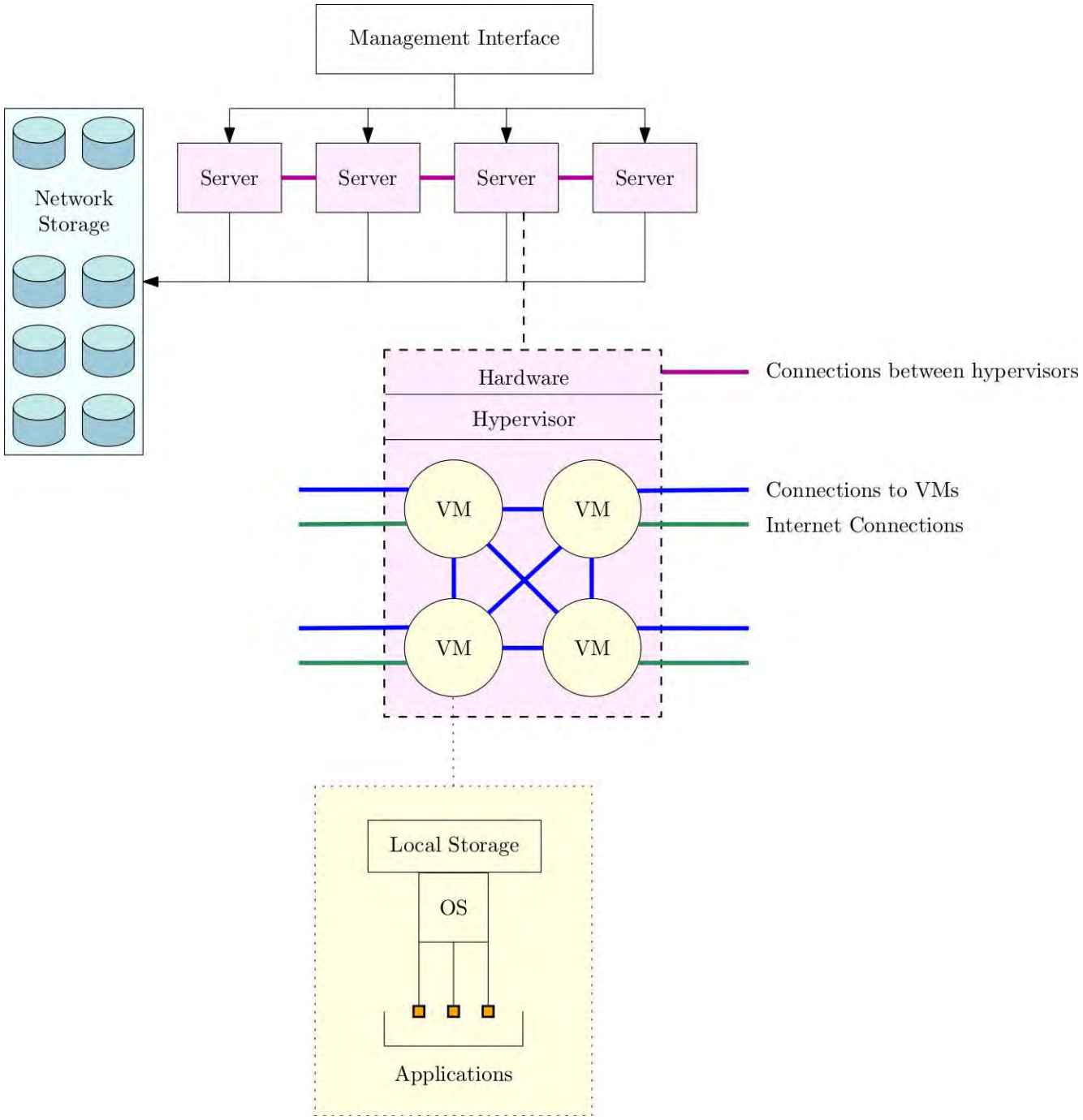


Fig. 1: The standard architecture of cloud computing infrastructure. Note this same infrastructure can be used to provide Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).

the consumer is given access to an EC2 "instance" (a VM) for a period of time to be used as a resource for whatever purpose the consumer wishes. Another example of IaaS would be Amazon's S3 service: the consumer is given access to low-latency data storage that is accessible from any location via the Internet.

With Platform as a Service (PaaS), the consumer has access to computational platforms including operating systems, programming language execution environments, databases, web servers, etc. These combined services are mainly used by developers who use the provided platform to run and test their software solutions on a cloud infrastructure without the overhead of maintaining the underlying software or hardware. Google App Engine [2] is an example of PaaS which is utilized for developing and hosting web applications within Google-managed data centers. The developed applications are sandboxed and run across multiple servers for testing. Amazon Web Services (AWS) Elastic Beanstalk [3] is another PaaS system where clients are able to deploy their created or acquired applications on a virtual machine (e.g., in the form of

a runnable jar) in order to test and deploy it. AWS Elastic Beanstalk is built on top of Amazon EC2, S3, and other parts of Amazon's IaaS offerings.

In the software as a service (SaaS) service model, the provider installs and operates application software on a cloud infrastructure. Clients may then access the software using a service-specific client software or a generic web browser interface. As with PaaS, SaaS providers are often consumers of IaaS. An example of this would be Dropbox. Dropbox allows clients to store their data and access it from any location via either the Dropbox website or the software one can install on their personal machine. Note that Dropbox has its software running overtop Amazon's S3 service for mass data storage [4]. Netflix is also a company that both provides and consumes cloud computing services. Netflix allows consumers to access movies and TV shows from any location via their website or installed application. While providing this service, Netflix layers their software and functionality atop Amazon Web Services [5]. Another example of SaaS is the Google search engine. Clients access their search engine through a standard web interface and are able to search the Internet for answers, solutions, etc. However, contrary to the traditional approach to cloud computing used by companies such as Dropbox and Netflix, the Google search engine uses its own infrastructure and does not employ VMs [6]. In other words, the Google search engine does not layer its software atop a cloud infrastructure that is already in place, such as Amazon EC2. Facebook also falls into the SaaS model, but follows the same vein as the Google search engine in that Facebook has defined, implemented, and uses its own infrastructure without utilizing VMs or third-party cloud services such as Amazon EC2 [7].

Thus while these different service models have different economic, administrative, and consumer experiences, they all share a common architecture that is typically something akin to that shown in Fig. 1. Variations include the use of local versus remote storage pools and the degree to which hypervisors are employed to host full operating systems or the use of other mechanisms, such as language-based virtual machines, to separate customers. For example, as we mentioned, Google and Facebook do not use virtual machines because users of their services—even services such as Google App Engine [8]—are only allowed to run certain types of software running in restricted environments. Most others in cloud computing, however, either provide or make use of collections of virtual machines connected to storage pools.

It should be noted that the instances of these models which we have detailed above are all examples of public clouds—clouds that are made for consumption and utilization by the general public—rather than private clouds which have been internally constructed for use by one company or organization. As we will see, this key difference enables several new kinds of attacks. Throughout the paper we utilize Amazon and its cloud services as the main example of a public cloud provider and platform. We do this as Amazon is the most popular platform at present for a public cloud. However, they have many prominent competitors including RackSpace, CloudSpace, and Microsoft Azure. While their technology stacks differ, significant overlap in functionality and increasing interface standardization means that customers can migrate between services and develop systems that span providers. Thus the attacks and defenses outlined in the paper, while taking Amazon as their main example, also apply to other public cloud providers. These attacks and defenses are also applicable to private clouds, but only to the extent that attackers can gain access to the cloud infrastructure.

III. Security and the Cloud

Cloud computing infrastructure is, in principle, subject to all of the threats that standard server computing infrastructure is. Web servers can be compromised with cross-site scripting vulnerabilities; databases are subject to SQL injection attacks; operating system kernels can be compromised by machine code injection. Here, however, we are concerned with ways in which cloud-based systems are different from traditional servers from a security perspective.

In the following sections, our focus is on attacks that only make sense in a cloud computing context, as these are the new risks that arise when transitioning to the cloud. We should note, however, that cloud-based systems potentially do have some security advantages. Cloud providers can automate and provide as a service many standard systems administration tasks such as backups, software patching, and network monitoring. Virtual machines may be "reinstalled" very quickly through automated provisioning, allowing virtual machines that have been compromised to be more easily replaced than servers running on raw hardware. The security state of virtual machines and their associated storage may also be monitored externally (outside the scope of the guest VM's potentially untrustworthy applications and operating system) for malware by scanning files and even having the hypervisor directly detect intrusions in running VMs using introspection techniques [9]. While these are potentially useful, they are also things that could be implemented outside the cloud. The attacks and defenses we discuss in the rest of this paper, however, are all unique to applications running in the cloud, particularly public clouds.

In the rest of this paper we make the following assumptions. We assume that cloud applications are run within virtual machines running on hypervisors with local storage and access to remote network storage as shown in Fig. 1. The target (victim) is assumed to have one or more VMs in the cloud. We assume that the attacker is either on the public Internet connecting to the targeted VM or that the attacker has a VM with the same cloud service as the targeted VM. For some attacks, we further assume that the attacker has a VM on the same host—running on the same hypervisor—as the target. In all of our scenarios, we assume that the target's software—their applications and operating system—are otherwise secure. Thus the attacker is primarily taking advantage of the fact that the target is making use of a cloud computing infrastructure.

IV. COLOCATION: DENIAL OF SERVICE

In a cloud infrastructure, CPU, RAM, disk, and network bandwidth resources are shared between users. As such, if an attacker consumes a large amount of resources, all other customers that share the same physical resources will notice a decrease in performance. If severe enough, this decrease constitutes a denial-of-service attack. Customer applications may be migrated to other part of the cloud infrastructure with less resource contention; however, even the largest of providers have finite resources.

Cloud providers employ a variety of strategies to partition resources in such a way that such denials of service—whether accidental or deliberate—are less likely. Providers such as Amazon divide their cloud into "availability zones" that are designed to fail independently. To maximize uptime, developers must replicate their applications in multiple zones and allow fail-over between them. Within data centers, networks are partitioned by routers and network-level quality-of-service mechanisms [10]. Hypervisors such as Xen implement "fair share" CPU schedulers that give at most a fixed portion of a node's CPU and I/O bandwidth [11].

These partitioning strategies are not perfect, however, and it is possible to cheat, allowing users to exceed their allocation. For example, hypervisor schedulers can be manipulated into misallocating CPU resources [12]. Even if a customer's virtual machine gets its allocated share of resources, it may not get them in a timely fashion, causing increases in network response latency. Such increases in latency can be particularly harmful for cloud-hosted web applications. Thus one area of research to further explore is in improving latency under load [13].

Another key defense strategy is economic: they charge for resources used. Providers use existing metrics such as peak network bandwidth and storage consumption to measure and charge customers. Where metrics were not so readily available, such as CPU resources, providers have created new ones: Amazon's EC2 compute unit (ECU), for example, is defined as the power of a 1.0−1.2 GHz 2007-era AMD Opteron or Intel Xeon Processor [14], [15]. While the consumer is utilizing cloud services, this metric is monitored, most typically by a VM Monitor. Once the consumer has used the resources that their SLA has provided for (i.e. once the consumer has expended the amount of cloud services that they have initially agreed to and paid for), they are seen to be in violation of their SLA with the cloud provider. The cloud provider then utilizes a gradual formula to determine how much the SLA has been violated, meaning that the more the SLA is violated over time, the larger the penalty for the consumer will be. This penalty is typically in the form of financial recompense. With this kind of metering in place, resource-based denial of service then often becomes a matter of fraud, either of evading the metering mechanisms or paying for services using stolen credentials (e.g., credit card numbers).

As providers get larger and better able to manage their resources using partitioning and economic mechanisms, pure colocation denial of service is becoming increasingly infeasible. Other kinds of attacks, however, are still very possible.

V. COLOCATION: BREACH OF CONFIDENTIALITY

With colocation-based breaches of confidentiality, attackers attempt to use colocation in order to compromise the confidentiality of a VM. Information about the data stored inside a VM can be inferred by noticing patterns of resource usage, particularly CPU usage. Such resource usage can be inferred through resource contention with a co-located attacker virtual machine.

For example, Ristenpart, et al. [16] outlined a series of attacks against the Amazon EC2 service. They would start up several instances (usually over 100 to gain the desired results) with the aim to hit a target. The first of these types of attacks is the gambler attack. This attack attempts to hit a target, any target, and compromise it. The second of these attacks is the sniper attack. During a sniper attack, attackers compromise a single, specific target. Once the attacker has chosen his intended victim(s), the attacker then attempts to influence the victim to react in a way that they can predict so that the attacker may extract information. Both of these attacks take advantage of the fact that many VMs will run on the same node (host). The creation and use of several hundred instances is meant to make it feasible for the attacker to land in the targeted (either arbitrarily or specifically) VM.

Attackers can use multiple ways to determine if they have landed on the target's node. One is a network-based strategy where simple IP scans are used to determine if the attacking instance and targeted instance share the same administrative IP address (e.g., the IP address of their Xen Dom0 instance). Another is to check whether there is a low latency network path with the target (i.e., whether packets can be exchanged with minimal transmission delay). Or, the attacker can check to see whether accesses to the target increase the rate of cache misses in the attacker's VM; if it does, then they are sharing hardware [16]. Once the attacker shares a node with the target, timing and cache interference effects between VMs can be further used to extract information from the target, such as OS information [17] and even cryptographic keys [16], [18].

There are a number of stages where defenses can help prevent these kinds of attacks. One is to prevent the attacker from sharing hardware. While this sort of protection can be gained by moving to a private cloud, even public cloud providers can give some of this type of protection by guaranteeing exclusive access to nodes (for an extra fee, of course). The provider can also randomize their VM distribution schemes to reduce the probability of attacker/target co-location. But if we assume the attacker will be running on the same node (as is likely for a gambler attack), then we must minimize potential communication channels between VMs.

The next step would be for cloud providers to block the side channels that attackers may exploit. There are three approaches that a cloud provider could take for this with regards to the cache-level attacks. The first of these is to guarantee exclusive access to CPU caches (L1, L2, or L3). If a

consumer has exclusive access to their caches, they can potentially detect hostile intrusions to their VMs via analysis of cache level noise [19]. If the consumer is barred exclusive access to the entire cache, they can be granted exclusive access to a portion of the cache through cache partitioning, for example through cache coloring [20]. In partitioning the cache, the amount of cache information overflow that may cause information leakage is greatly reduced as the attacker may no longer monitor what is being ejected from or altered within the cache by other VMs. Finally, if the consumer is not guaranteed exclusive access to the cache in any way, they may then be able to monitor the cache to determine if an attacker is attempting to extract any information. One cache-level attack strategy involves priming the cache with a large amount of temporary files. Such patterns of of malicious behavior can be detected through VM introspection [21]. Also, cryptographic timing attacks can also be mitigated by reducing the precision of the system clock, as these attacks require very precise timing [22].

VI. Data Availability and Integrity

Here we consider the following problem: how can a customer trust that a provider has the data they are supposed to be storing? Specifically, how does a customer know whether their data is accessible and has not been corrupted? Currently, cloud providers only guarantee uptime in their service level agreements; they do not explicitly guarantee data integrity or availability [23]. As such, cloud providers are under no obligation to prevent or notify the consumer of data corruption or loss of data availability.

The customer can, of course, verify data accessibility and integrity by manually accessing all remotely stored information. Computational and bandwidth constraints, however, make conventional implementations of such operations prohibitive in most contexts. Research into this area focuses on ways to make customer checks of data more feasible.

For example, the protection mechanism of the High Availability and Integrity Layer (HAIL) [24] attempts to do this by implementing similar functionality to Redundant Array of Independent Disks (RAID), in that data is mapped onto multiple (virtual) disk drives using a combination of two cryptographic functions–Proof-of-Retrievability [25] (POR) and Proof of Data Possession [26] (PDP). POR is a cryptographic function meant to enable a prover (the cloud provider) to demonstrate to a verifier (the consumer of cloud services) that a certain file is retrievable. This is done with the use of a small checksum, giving a high efficiency benefit as only a very small amount of data, not an entire file, needs to be transmitted. PDP is meant to show that a file stored in the cloud has not been altered or modified and that the consumer has access to said file without the need to fully download it.

While the POR and PDP functions of HAIL can greatly reduce the bandwidth required to verify data availability and integrity, they both have high enough computational complexity that they are not feasible to be implemented on today's systems. Fortunately mechanisms for integrity and availability monitoring of cloud providers is an active area of research [24], [25], [27]. Practical, deployed solutions to customer checking of data, however, still remain to be developed.

VII. Data Confidentiality

While consumers of cloud services desire cloud providers to both store and serve their data, they do not necessarily want the cloud providers to have free access to their data as this would be a breach in confidentiality. Yet today there are currently no default methods in place to prevent cloud providers from having free and ready access to the data they are storing and serving. Malicious providers of cloud services could freely peruse the data they are given access to by clients as, by default, all data is stored in the clear.

The natural solution is to encrypt cloud resident data. Simple encryption, however, is not so straightforward to implement. One reason is that cloud-based virtual machines are generally used to process cloud-resident data. They must have access to the data, so even if the data is encrypted the cloud provider also, implicitly, has the key to the data (in the form of the running virtual machines). Thus, encrypted storage for completely cloud-based systems has inherently limited benefits.

Another problem with encrypted storage in the cloud is that naive implementations of file-level or block-level encryption are all subject to traffic analysis. In other words, data access patterns themselves can breach confidentiality even if all of the data being accessed is encrypted. To prevent traffic analysis, data from different files have to be mixed together in terms of both reads and writes in such a way as to confound traffic analysis without incurring too much overhead.

Another aspect of this traffic analysis problem is that file metadata, in addition to file data, must be encrypted to prevent monitoring of specific users or groups. This problem is particularly significant for enterprises wanting to give selective access to their cloud-based data. Multiple privacy-preserving access control mechanisms have been proposed. An example would be the system proposed by Raykova et al. [28]—a double layered Access Control List (ACL) would be in place whereby one layer would specify what files a cloud provider should and should not have access to and a secondary layer for the user, when their VM is up and running, to provide and specify fine grain access. The general idea behind this being that, just because user data is being stored on the cloud does not mean that the cloud provider should have access to said data, nor should they have knowledge of who is able to access the data at all. Furthermore, as the user data is now stored on a public cloud, not a private VM or network, the ACL mechanisms need to take into account the large number of users accessing the cloud services and, as such, should provide much finer grained controls. Yu et al. [29] and Wang et al. [30] have also outlined ACL systems that could be applied to the cloud. These proposals do not appear to currently be mature enough, however, to be implemented on current systems. We see there is a big opportunity in this area

for solutions that provide sufficient confidentiality while being practical for current providers, applications, and customers to use.

VIII. Infrastructure Compromise

Infrastructure compromise is the most unexplored threat area in cloud security. However, it is also the attack with the highest amount of payoff: If successful, the attacker gains a level of privilege akin to attaining root access on a machine. The weak point exploited by the attacker is a management interface of the provider (internal or external) rather than a direct attack against the cloud infrastructure. Even though it is the provider's resources that are exploited, the attack affects consumers as well.

Attacks against the management interface of cloud services are mostly unexplored. Amazon's cloud service provides a Simple Object Access Protocol (SOAP) and REST-based interface, with the SOAP interface being defined by an XML schema [31]. One aspect of the attack outlined by Somorovsky et al. [32] focuses on the SOAP-based aspect of this interface where a developer may post a SOAP message with XML signatures. To exploit, all the attacker needs is a valid, signed SOAP message. Such messages can be easy to obtain; for example, developers tend to include them in public message postings in order to aid with debugging. The attacker may then use these messages and their keys to forge a new, malicious message. With this forged message the attacker can trick the host, Amazon, into thinking that the attacker is a legitimate administrative user for that domain, hence giving the attacker complete control over that domain. This attack was first described by McIntosh and Austel in 2005 [33].

It should be noted that Gruschka and Iacono [34] originally proposed a similar attack, though their version of the attack had the disadvantage of being time sensitive. However, Somorovsky et al. [32] later solved the time sensitivity problem, rendering the attack more feasible.

So far, we have encountered no research targeted at preventing these types of attacks–compromising and maliciously using the cloud management interfaces. While specific vulnerabilities can be addressed by software updates, such fixes can only be implemented when vulnerabilities are disclosed. Zero-day exploits are particularly dangerous in the context of cloud infrastructure compromise given the leverage an attacker can gain from a single successful attack. An open area of research, then, is how to harden or otherwise defend against attacks on previously unknown vulnerabilities in cloud management interfaces.

IX. Related Work

Several reviews have already been performed regarding both to the cloud infrastructure and its current state of security. Almorsy et al. [35] and break cloud infrastructure down into the component service models Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) (See Section II). Chen et al. [36] go on to review whether or not there are any new threats or protections within cloud security. They make the point that some of the threats seen so far within the cloud infrastructure are new only in the sense that they are being seen in the cloud computing model rather being used to target single machines (e.g. installation of malware). Lastly, Lombardi et al. [37] give an overview of the current threat model of the cloud, providing both a list of attacks and the requirements for these attacks. Following their definition of the current threat model of the cloud, they present a detailed framework to categorize the attacks (our attack categorization is similar, but not identical, to theirs). While these reviews have made notable contributions to analyzing the current state of cloud security, none of them cover both cloud attacks and defenses.

X. Discussion

The fundamental issue with the move to the public cloud is that "hardware" is now much less trustworthy than before. Attackers in the cloud can run their code on the same hardware as the victim without bypassing any access controls; instead, they just need to manipulate the cloud provider so as to share resources with the target. This below-the-operating-system level of vulnerabilities is also strictly additive: all of the old vulnerabilities in operating systems kernels, system libraries, applications, and user behavior are still present.

The issues of co-located denial of service are being reasonably well addressed today, simply because this problem is fundamental to the business model of public cloud providers. If customers get poor service, they will take their business elsewhere. Co-located confidentiality breaches, particularly through cache attacks, as we have shown are being actively studied in the research literature. However, these attacks are all complex and are likely only to work against a small subset of virtual machine workloads where cryptographic operations take place very frequently.

While there is less work on cloud-specific data integrity, availability, and confidentiality issues, previous practice with non-virtualized resources can address many of these issues, at least partially. At the end of the day, though, placing data in long-term storage in the cloud is an act of trust. As we have shown, there are some technical solutions that can help reduce the amount of trust that must be placed in the cloud. In practice, however, these problems are more often being addressed through contracts and reputations. While such social arrangements might appear to be problematic, the scale at which cloud providers operate allows them to implement best practices for data management. As such, they may be more trustworthy than local storage for most customers.

Nevertheless, cloud providers are very tempting targets for attackers, particularly sophisticated ones. If they can get control of a cloud provider's infrastructure, whether through external or internal interfaces, they can control the fates of thousands of cloud customers and millions of individual users. The automated mechanisms that allow resources to be allocated and de-allocated on demand could become a huge force multiplier in the wrong hands. Research into how to harden this infrastructure is difficult as much of the current technology is proprietary and is controlled by relatively few

companies. The cloud technology stack, however, is becoming more standardized with initiatives such as OpenStack [38]. No matter how hard it is to do, such research is needed so we can, at a bare minimum, better understand the risks we are running with the ongoing migration of our computational lives to the cloud.

REFERENCES

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," NIST special publication 800-145, September 2011.

[2] Google Inc., "Google app engine – google developers," https://developers.google.com/ appengine/, accessed on March 24, 2013.

[3] Amazon Inc., "AWS elastic beanstalk," https://aws.amazon.com/elasticbeanstalk/, accessed on March 24, 2013.

[4] Dropbox, "Dropbox help - where does Dropbox store everyone's data? " https://www.dropbox.com/ help/7/en, accessed on March 15, 2013.

[5] J. Ciancutti, "Four Reasons We Choose Amazon's Cloud as Our Computing Platform," http://techblog.netflix.com/2010/12/four-reasons-we-choose-amazons-cloud-as.html, accessed on March 15, 2013.

[6] "Google site search solutions," http://www.google.com/enterprise/pdf/google_site_search_solutions.pdf, accessed on March 15, 2013.

[7] P. Peacock, "Web-based vs cloud-based," http://thecloudandme.com/2010/03/18/web-based-vs-cloud-based/, accessed on March 15, 2013.

[8] "What is (isn't) Google App Engine, https://developers.google.com/appengine/training/ intro/whatisgae, accessed on March 15, 2013.

[9] M. Christodorescu, R. Sailer, D. Schales, D. Sgandurra, and D. Zamboni, "Cloud security is not (just) virtualization security: a short paper," in Proceedings of the 2009 ACM workshop on Cloud computing security.

[10] A. Shieh, S. Kandula, A. Greenberg, and C. Kim, "Seawall: performance isolation for cloud datacenter networks," in Proceedings of the 2nd USENIX conference on Hot topics in cloud computing, 2010.

[11] D. Ongaro, A. L. Cox, and S. Rixner, "Scheduling I/O in virtual machine monitors," in Proceedings of the fourth ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, 2008.

[12] F. Zhou, M Goel, P. Desnoyers, R. Sundaram, "Scheduler Vulnerabilities and Coordinated Attacks in Cloud Computing," 10th IEEE International Symposium on Network Computing and Applications (NCA), 25-27 Aug. 2011.

[13] S. Agarwal, J. Dunagan, N. Jain, S. Saroiu, A. Wolman, and H. Bhogan, "Volley: Automated data placement for geo-distributed cloud services," in Proceedings of the 7th USENIX conference on Networked systems design and implementation, 2010.

[14] Í. Goiri, F. Julià, J. Fitó, M. Macías, and J. Guitart, "Resource-level QoS metric for cpu-based guarantees in cloud providers," Proceedings of the 7th International Workshop on Economics of Grids, Clouds, Systems, and Services (GECON 2010). LNCS Vol. 6296, Springer, 2010.

[15] Amazon Inc., "Amazon EC2 SLA," https://aws.amazon.com/ec2-sla/, accessed on December 12, 2012.

[16] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds," in Proceedings of the 16th ACM conference on Computer and communications security, 2009.

[17] R. Owens and W. Wang, "Non-interactive os fingerprinting through memory de-duplication technique in virtual machines," in 2011 IEEE 30th International Performance Computing and Communications Conference (IPCCC), 2011.

[18] Y. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Cross-VM side channels and their use to extract private keys," in Proceedings of the 2012 ACM conference on Computer and communications security, 2012.

[19] Y. Zhang, A. Juels, A. Oprea, and M. Reiter, "Homealone: Co-residency detection in the cloud via side-channel analysis," in 2011 IEEE Symposium on Security and Privacy (SP).

[20] J. Shi, X. Song, H. Chen, and B. Zang, "Limiting cache-based side-channel in multi-tenant cloud using dynamic page coloring," in 2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W), 2011.

[21] A. Srivastava, K. Singh, and J. Giffin, "Secure observation of kernel behavior," 2008.

[22] B. C. Vattikonda, S. Das, and H. Shacham, "Eliminating fine grained timers in Xen," in Proceedings of the 3rd ACM workshop on Cloud computing security workshop (CCSW '11). 2011.

[23] Amazon Inc., "Amazon customer agreement," https://aws.amazon.com/agreement/, accessed on December 12, 2012.

[24] K. Bowers, A. Juels, and A. Oprea, "HAIL: a high-availability and integrity layer for cloud storage," in Proc. of the 16th ACM conference on Computer and communications security, 2009.

[25] A. Juels and B. Kaliski Jr, "Pors: Proofs of retrievability for large files," in Proceedings of the 14th ACM conference on Computer and communications security, 2007.

[26] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proceedings of the 14th ACM conference on Computer and communications security, 2007.

[27] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in INFOCOM, 2010 Proceedings. IEEE, 2010.

[28] M. Raykova, H. Zhao, and S. Bellovin, "Privacy enhanced access control for outsourced data sharing," Financial Cryptography and Data Security, LNCS Vol. 7397, Springer, 2012.

[29] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in INFOCOM, 2010 Proceedings. IEEE, 2010.

[30] G. Wang, Q. Liu, and J. Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," in Proceedings of the 17th ACM conference on Computer and communications security (CCS '10), 2010.

[31] S. Gajek, M. Jensen, L. Liao, and J. Schwenk, "Analysis of signature wrapping attacks and countermeasures," in IEEE International Conference on Web Services (ICWS), 2009.

[32] J. Somorovsky, M. Heiderich, M. Jensen, J. Schwenk, N. Gruschka, and L. Lo Iacono, "All your clouds are belong to us: security analysis of cloud management interfaces," in Proceedings of the 3rd ACM workshop on Cloud computing security workshop, 2011.

[33] M. McIntosh and P. Austel, "XML signature element wrapping attacks and countermeasures," in Proceedings of the 2005 workshop on Secure web services. ACM, 2005.

[34] N. Gruschka and L. L. Iacono, "Vulnerable cloud: Soap message security validation revisited," in 2009 IEEE International Conference on Web Services (ICWS 2009).

[35] M. Almorsy, J. Grundy, and I. Müller, "An analysis of the cloud computing security problem," in Proc. of the 2010 Asia Pacific Cloud Workshop, Colocated with APSEC2010, Sydney, Australia, 2010.

[36] Y. Chen, V. Paxson, and R. Katz, "What's new about cloud computing security? " University of California, Berkeley Report No. UCB/EECS-2010-5, January 2010.

[37] F. Lombardi and R. Di Pietro, "Secure virtualization for cloud computing," Journal of Network and Computer Applications, vol. 34, no. 4, pp. 1113–1122, 2011.

[38] OpenStack Foundation, "Open stack," http://www.openstack.org/, accessed on March 25, 2013.

Presentation: Data Breach Reporting Preparation: An Analysis of Practice

Ernst Bekkering, Associate Professor
Department of Information Systems, Northeastern State University, OK

DATA breaches continue to occur at an alarming rate. Unauthorized access to and loss of data with financial and privacy impact occurs across public and private sectors, industries, and organizations large and small. Exposure to data breaches is becoming more an issue of when, not if, it will occur. Depending on the magnitude of breach, type of data, industry, and location, incidents need to be reported to potential victims, outside agencies, and/or law enforcement. Formulating Incident Response Policies and designating Incident Response Teams are essential tools for managing potential future incidents. This presentation discusses a study of the level of preparation across organizations. Preliminary results will be presented at the conference.

Mobile Security and Vulnerability Exploitation as a Flipped Classroom Security Curriculum

Richard P. Mislan, Ph.D. and Tae Oh, Ph.D., *Senior IEEE*

***Abstract*— While mobile devices have become essential to the daily social fabric of our lives they have also become a popular platform to exploit. The security of mobile devices is a growing necessity, yet many in our population are woefully inexperienced in providing proper security measures. In an effort to address this need, this paper discusses the development of a unique classroom model based on the flipped classroom that provides a repository website of integrated course resources and virtualized laboratories for the education of "Mobile Device Security and Vulnerability Exploitation." Given the specific needs of this type of mobile security modeling, it is imperative that the students participate in a secured laboratory setting. To meet this necessity, the development of a website repository and the inclusion of video lectures, presentations, and virtualized laboratory exercises specific to the course is proposed.**

***Index Terms*— Mobile Device, Exploitation, Security, Curriculum, and Flipped Model**

I. Introduction

AS THE proliferation of mobile devices continues to rise in the personal and professional world, there is a growing need for better understanding and awareness of mobile device security and forensics. Currently in the United States, there are more wireless mobile devices connected than there are people (331.6M/311.5 at 106%) and the number of devices is projected to continue to rise [4]. As the ultimate human computer interface of the 21st century, the mobile device allows almost everyone to do almost everything, almost everywhere they go. Consolidating many bleeding edge technologies into a single unit, the mobile device allows us to transcend space and time through multiple communication, transaction and entertainment tools.

As mobile devices have become essential to the social fabric of our lives, the past year has seen a 2,180% increase in unique malware variant attacks [5]. With always-connected capabilities, mobile devices have become the ultimate access point to a person's most private and personal information. Once a device has been maliciously compromised, personal data is exposed, leaving the owner violated and picking up the pieces. Blurring the line between personal and professional is the critical issue of Bring Your Own Device (BYOD), which impacts all sectors.

As important as mobile communications tools have become, the security of mobile devices is still overlooked by our society. The authors, as members of a highly technical community and as users of such devices, feel a strong sense of urgency to resolve the issues related to mobile device vulnerabilities through increased awareness and mitigation [6], [7].

Therefore, the authors have decided to design a course to introduce the topics of mobile device vulnerability exploitation with the goal of understanding existing mobile exploits to build better defenses. The breadth of academic research in mobile device vulnerability exploitation has not yet matured, leaving this field a ripe opportunity for exploration and discovery.

This paper introduces the meaning and purpose of the flipped classroom model and discusses why this model is being selected as a development framework for this course. The paper then describes how the mobile device vulnerability exploitation course is designed, including class materials, lab exercises, an online repository, complementary courses and other materials. Finally, the paper concludes with expectations, future work, and other opportunities for discussion.

II. Flipped Classroom Model

While not a new teaching model, the flipped classroom is rapidly becoming a popular choice amongst innovative educators [1]. The flipped classroom model inverts traditional teaching methods by delivering instruction outside of the class using online resources, thus shifting traditional homework exercises into the classroom [9]. Before attending a class session, students are assigned online lectures to view at home or any other location they may be, and to communicate with fellow students and faculty via online discussions [3]. During the classroom session, student engagement occurs by working through small problem sets and exercises, assimilating previously learned information, and creating new ideas within the provided time with the guidance and/or assistance from the instructor.

This paragraph of the first footnote will contain the date on which you submitted your paper for review. It will also contain support information, including sponsor and financial support acknowledgment. For example, "This work was supported in part by the U.S. Department of Commerce under Grant BS123456".

The next few paragraphs should contain the authors' current affiliations, including current address and e-mail. For example, F. A. Author is with the National Institute of Standards and Technology, Boulder, CO 80305 USA (e-mail: author@ boulder.nist.gov).

S. B. Author, Jr., was with Rice University, Houston, TX 77005 USA. He is now with the Department of Physics, Colorado State University, Fort Collins, CO 80523 USA (e-mail: author@lamar.colostate.edu).

T. C. Author is with the Electrical Engineering Department, University of Colorado, Boulder, CO 80309 USA, on leave from the National Research Institute for Metals, Tsukuba, Japan (e-mail: author@nrim.go.jp).

The flipped classroom was created to address the lack of achieving learning outcomes, and to increase the limited concept engagement of the traditional classroom model [9]. In this case, the authors will be using the flipped classroom model for two reasons: 1) to provide students with an immediate immersion into the field through an online repository of readings, presentations, and videos related to mobile device security and vulnerability exploitation, and 2) to maximize the classroom time for supportive application of theories and practices of mobile device securitya nd vulnerability exploitation and various hands-on exercises.

As published in numerous articles and journals, the flipped classroom promotes a stronger student/teacher relationship and creates a collaborative learning environment in the classroom. By shifting the preparation of the student to an online component, the in-class laboratory exercises will be much more meaningful and provide time for collaboration and curiosity.

III. Course Objectives

To increase student knowledge of mobile device security and vulnerability exploitation, the following objectives are defined:

Objective 1: Development of lectures

In developing the lectures specific to the subject area, videos will be 5-7 minutes in length. Major topics will be broken into much smaller "digestible" topics. These will be built for preparing the student for the laboratory exercises to be conducted during class.

Objective 2: Development of an online repository of integrated course resources

To support the development of the flipped classroom, an online repository of integrated course resources and virtualized laboratories will be built. This website will categorize the information related to each weekly topic. Each week will have resources categorized as: "Laboratories," "Presentations," "Research," "Videos," and "Assessments." These will be used both in class and out of class.

Objective 3: Development of virtualized laboratories

The unique situation of mobile exploitation is that most laboratory exercises must be performed off of the mobile carrier network. To that end, the laboratory exercises must either be virtualized in an online simulation or carried out through a privately established low-power network. The former option is much less expensive and much more controllable. For that reason alone, it is necessary to establish virtualized laboratory exercises.

Objective 4: Development of assessment tools

In an effort to continually improve the process of flipped classroom learning, students will be assessed through a variety of pre- and post- type of assessments. Online resources will survey students' knowledge before viewing video lectures or presentations. These same online resources will survey students' gained knowledge after viewing the presentation. Short answer questions related to the content as well as multiple choice and true-false type assessment tools will be employed to validate strengths and weaknesses of the provided lectures and/or presentations.

To finalize the project, all curricula, lab information, key findings, and pilot results will be transparently shared through presentation, reports, and the specific curriculum website. Through the deployment of this project, students will obtain critical skills and knowledge directly related to security and vulnerability exploitation in a mobile environment. In the most basic terms, through its curriculum, academic sharing of materials, and collaborative research opportunities, this project will increase the number of students capable and qualified to teach others about mobile device security and vulnerability exploitation. By extension, this will also increase student learning in this critical area, increasing the workforce pipeline of qualified graduates to support and secure our mobile cyber infrastructure.

IV. Methodology

To create a standard expectation for the course curriculum, the objectives for each course will be aligned to the stated outcomes used for accreditation by ABET/Middle States. Once these objectives are defined, weekly course materials for mobile device vulnerability exploitation topics will be built. Through this course, students will learn about a) mobile communications technologies, b) mobile operating systems, c) current threats, d) mobile malware, vulnerabilities, and exploits, and e) code and application analysis tools and techniques. For many of these topics, the flipped classroom model and hands-on lab exercises will reinforce the students' learning.

Through the use of a flipped classroom curriculum and various collaboration opportunities, this effort will enhance and strengthen the capabilities of RIT students and faculty. Ongoing research and development exercises of mobile device vulnerability exploitation will be created and developed to encourage participation of other faculty and students as well as other Universities [11].

V. Course Topics and Materials

For the 15 week semester, there will be 15 topics covered (See Table I). The first week will be an introduction to the topic of mobile device vulnerability exploitation. This will be followed by a two weeks of the technical and social topics of mobile devices. The next two topics will cover the data storage methodologies of application data. After that, the students will be introduced to current mobile exploits, and then will explore the popular smartphone operating systems and the specific exploit that exists for each. The course will end with a short introduction to advanced mobile device security and exploitation techniques, somewhat preparing them for their next class.

TABLE I
COURSE OUTLINE FOR MOBILE SECURITY AND VULNERABILITY EXPLOITATION

(i)	Topics
1	Introduction to Mobile Devices, History/Appreciation
2	Network Stacks and Mobile Firmware
3	OS's: Android, Apple, Blackberry, Windows
4	Forensics: Procedures/Principles, Tools, Techniques
5	Forensics: Basic Phone

6	Forensics: Smartphones
7	Malware: Introduction/History/Appreciation
8	Mods: Jailbreak, Root, Unlock
9	Apps: Security and Vulnerabilities
10	Attacks: Hardware and Device
11	Attacks: Software
12	Attacks: User Layer
13	Strategy: Threat Preparedness and Mitigation
14	Strategy: Threat Response and Recovery
15	Legal/Ethical/Moral Issues

VI. LABS

During the semester, the students will use hands-on exercises that will allow them to explore the intricacies of the mobile application environment as shown in Table II. Each lab activity will include previously imaged, specific operating system mobile devices for student exploration and examination. Lab exercises will provide foundational focus on:

- The identification of operating system specific user data storage locations.
- The identification of operating system specific user data storage methodologies.
- The identification of common application data storage methodologies.
- The usage of common mobile exploits for specific operating systems.
- The effects of common mobile exploits on specific operating systems.

The labs will be spread throughout the semester and will be offered after the topic has already been presented and discussed online. These exercises will be designed to promote various ways to effectively increase knowledge and comprehension of the concepts, especially to those students who are kinesthetic learners.

TABLE II
COURSE LAB FOR MOBILE SECURITY AND VULNERABILITY EXPLOITATION

Topics
Introduction to Mobile Devices
Mobile Device Forensics
Mobile Device Security
Mobile Attacks Vectors Classes and Attack Models
Mobile Strategy, Policy and Risk Management
Legal/Ethical/Moral issues

VII. DIGITAL SMARTPHONE CORPUS

As noted by other security and forensics researchers, "real data is often unsuitable for education purposes" in large part because of confidential information found in digital devices [2]. Most attempts at this often end up as insufficiently realistic. Usually, in these attempts to mimic real-world devices, the data sets become more complex than is necessary. Given those challenges, there is still a need for the development of a corpus of smartphone device images for these courses. To that end, a corpus of operating system specific devices will be created to demonstrate and highlight:

- data storage locations
- data storage methodologies
- attack vectors for exploiting
- the effects of exploits

The idea for a corpus of digital data is not new. Currently, there are two small public corpora of non-smartphone mobile devices. Created by Simson Garfinkel and supported in part by NSF Grant DUE-0919593, the Real Data Corpus (RDC) is a collection of raw data extracted from data-carrying devices that were purchased on the secondary market around the world [2], [13]. The other corpus is the Computer Forensic Reference Data Sets (CFReDS) for digital evidence provided by the National Institute for Standards and Technology [8]. Both of these reference data sets provide documented sets of data from hard drives, cell phones, USB memory sticks, and other data-carrying devices. Neither of them currently house any smartphone images.

VIII. ONLINE REPOSITORY

We are building an online repository of information for students to immerse themselves into this burgeoning field of study. This repository will include curriculum, a virtual lab, an assessment system, and a faculty/student outreach vehicle for dissemination and collaboration. Borrowing from the previous successful efforts of other network security educational projects such as the NSF funded SEED program from Syracuse University[1] and the related SWEET program from Pace University[2], and the Security Injection program from Towson University[3], this tool will be used to share curriculum ideas, the corpora of smartphone images, and online resources for other educational opportunities [12]. It will also provide support for existing, ongoing, and future mobile exploitation research, collaboration, and dissemination. Ideally, access to these materials will directly benefit the educational community in mobile security and vulnerability exploitation, as well as the general population as it pertains to securing our world of mobile devices.

IX. COURSE EVALUATION

The evaluation of this project will be conducted through a variety of formative and summative tools. The process evaluation tools that will be used to determine the effectiveness of the flipped classroom model and its correlated laboratory exercises include following measurement techniques.

A. Quizzes-Measure Understanding of Classroom Materials

Before class starts, each student will take short quizzes from the previous-to-classroom viewed materials. The results of the quiz will indicate the comprehension level of the students. If the score is low, the instructor will determine reasons for the low grade and update the course materials to adjust the course

[1] http://www.cis.syr.edu/~wedu/seed/
[2] http://csis.pace.edu/~lchen/sweet/
[3] http://triton.towson.edu/~cssecinj/secinj/

delivery methods.

B. Surveys - Measure Video Lecture Effectiveness

Before and after each instructor prepared video lecture related to course concepts, each student will take a short online survey to assess pre- and post- knowledge and comprehension. Short answer questions related to the content as well as multiple choice and true-false type assessment tools will be employed to validate strengths and weaknesses of the provided lectures and/or presentations. If the average scores are too low or too high, the instructor will determine reasons for the low or high scores and update the course materials to improve the course delivery methods.

C. Laboratory Exercises - Measure Hands-on Learning Effectiveness

The laboratory exercises will apply the concepts from lectures and video demonstrations and the grade from laboratory report will indicate the students' comprehension, understanding and learning level.

D. Course Evaluation - Measure Effectiveness of this Course

The third measure will from the student feedback about this course. The instructor will be very interested in their interest level and learning effect. Collecting student's ideas and suggestions will improve the course. The progress will be closely monitored through classroom discussion as well as online discussion boards from the course website.

At the end of the semester, the course will be evaluated and restructured as necessary. Not all measurements are required to be implemented in the course and it's up to the instructor to select what measurements to be used and how often they are needed to meet the goals and purpose of the course.

X. Complementary Courses

The authors are also developing complementary classes in mobile device security and vulnerability exploitation for advanced studies and a capstone class for the advanced learner to test their abilities in exploiting and securing this trend in mobile devices. Brief description of those courses is described below.

A. Mobile Device Security

This course will introduce students to the various technologies employed in mobile device security. Emphasis will be placed on evaluating different types of mobile device malware and applications to determine the type of access and information disclosure threats that they represent [9]. Different types of malware detection solutions will also be identified and reviewed.

B. Mobile Device Vulnerability Exploitation

This course will introduce students to the various types of exploits available to attack mobile devices. Emphasis will be placed on evaluating different types of mobile device exploits and tools to determine the types of access and information available through each type of attack.

C. Capstone: Mobile Device Security and Vulnerability Exploitation

This course will further explore the various systems and technologies employed by mobile devices. Emphasis will be placed on the forensic identification of threats and vulnerabilities of specific mobile device operating systems and technologies and the application of and techniques for mobile device hardening.

XI. Current Status

As part of the curriculum development, funding was sought out through the Rochester Institute of Technology and was awarded through the Provost's office. Funds will go to towards the development of the course website and repository as well as to current mobile devices for lab work. To date, the website, curriculum, and repository is being built and hosted at MobiSploit.com. All information regarding development and progress on the curriculum can be viewed at the website.

Interactive labs for mobile device security are being created to instruct students how to evaluate and examine the behavior of the malware. The labs also teach students how to perform dynamic malware analysis and teach different analysis techniques in the malware analysis methods. Several tools will be introduced for the labs as well. Currently, there are four different labs are developed. The first lab is a brief introduction of Android emulator and introduces steps for setting up the testing environment. The second lab introduces SMS malware and the students can evaluate the behavior of the malware using different evaluation tools. The third and fourth labs introduce more sophisticated malware such as Trojan and botnets. Additional lab materials are needed and will be developed to support the understanding of mobile malware and Android architecture.

XII. Conclusion

Almost every student has a mobile phone; however, many of these students have no foundational understanding of mobile security and the weaknesses that lie within these devices. To that end, it is the authors' intent to develop a new course curriculum using the flipped classroom in mobile device vulnerability exploitation. This curriculum model will emphasize a hands-on approach by providing virtualized laboratory exercises using mobile device images and related mobile device emulator tools.

The core philosophy of the flipped classroom is that the authors encourage students to find and seek information and answers, thus promoting out of box thinking. The greatest benefit of this model is that students will learn from their investigative research and from each other through online presentations, discussions, and lab exercises. By implementing the flipped classroom model with virtual lab exercises, students will gain a practical level of knowledge about mobile device vulnerability exploitation, while the hands-on experiences will promote curiosity while producing better prepared, security minded students.

Sun Tzu, the ancient Chinese warrior taught his men to "know your enemy" before going into battle. His wisdom suggested that if "you know your enemy and know yourself, you need not fear the result of a hundred battles." But he also

warned, "If you know yourself but not the enemy, for every victory gained you will also suffer a defeat." The authors are taking this wisdom to heart as we develop this new curriculum. Not only are we studying, teaching, and researching the arts and methodologies of security, but we are also studying, teaching, and researching the vulnerabilities and exploitations of these mobile devices. As we prepare our students for their eventual future, it is critical that they know the vulnerabilities and exploits of the mobile devices as well as they know the methods of security.

As the authors continue to develop these courses, other more specific papers will be submitted in the future regarding the results of the flipped classroom, the usage of the online repository, and the development of the other complementary courses.

REFERENCES

[1] M. Becher, F.C. Freiling, J. Hoffmann, T.Holz, S. Uellenbeck, C. Wolf, "Mobile security catching up? Revealing the nuts and bolts of the security of mobile devices," 2011 IEEE Symposium on Security and Privacy. M. DeFour, "New 'flipped classroom' learning model catching on in Wisconsin schools," McClatchy - Tribune Business News, Feb. 2013.

[2] S. Garfinkel, (n.d.). Real data corpus. [Online]. Available: http://digitalcorpora.org/corpora/disk-images/rdc-faq

[3] K. Janz, K. Graetz, and C. Kjorlien, "Building collaborative technology learning environments," in Proceedings of the ACM SIGUCCS 40th annual conference on Special interest group on university and college computing services, New York, NY, USA, 2012, pp. 121–126.

[4] L. Liu, X. Zhang, G. Yan, and S. Chen, "Exploitation and threat analysis of open mobile devices," in Proceedings of the 5th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, New York, NY, USA, 2009, pp. 20–29.

[5] Y. Lv, D. Lymberopoulos, and Q. Wu, "An exploration of ranking heuristics in mobile local search," in Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, 2012, pp. 295–304.

[6] R. Mahmood, N. Esfahani, T. Kacem, N. Mirzaei, S. Malek, and A. Stavrou, "A whitebox approach for automated security testing of Android applications on the cloud," in 2012 7th International Workshop on Automation of Software Test (AST), June, pp. 22–28.

[7] S. Moran, "Security for mobile ATE applications," in 2012 IEEE AUTOTESTCON, Sept., pp. 204–208.

[8] National Institute for Standards and Technology Computer Forensic Reference Data Sets. [Online]. Available: http://www.cfreds.nist.gov/

[9] B. Starzee, "'Flipped classroom' model leaps to Long Island," Long Island Business News, Apr. 2012.

[10] A. Stavrou, J. Voas, T. Karygiannis, and S. Quirolgico, "Building Security into Off-the-Shelf Smartphones," Computer, vol. 45, no. 2, pp. 82–84, Feb.

[11] V. Tirronen and V. Isomöttönen, "On the design of effective learning materials for supporting self-directed learning of programming," in Proceedings of the 12th Koli Calling International Conference on Computing Education Research, New York, NY, USA, 2012, pp. 74–82.

[12] M. Vorsino, "Teachers explore 'flipped' class," Honolulu Star - Advertiser, Jul. 2012.

[13] K. Woods, C. Lee, S. Garfinkel, D. Dittrich, A. Russell, K. Kearton (2011). Creating realistic corpora for security and forensic education. ADFSL Conference on Digital Forensics, Security and Law, 2011. p. 123-134. [Online]. Available: http://simson.net/clips/academic/2011.ADFSL.Corpora.pdf

Android Malware Behaviors for Android Platform Using Interactive Labs

Colin Szost, Kriti Sharma, Tae Oh, Willaim Stackpole and Richard P. Mislan

***Abstract*— Android is a fairly new operating system launched by Google in October 2008; it has been gaining market and growing popularity ever since. Alongside the growth of the operating system, malware for Android has increased tremendously. Currently, few strategies are available to identify and detect malware on this platform. Users are generally naïve and are fooled into downloading apps, which may be harmful or malicious. They do not have enough experience with either the platform or the apps to know what can be malicious and what is genuine. Thus, the goal of the authors is to create a series of interactive labs that instruct students with little experience with the Android platform or dynamic malware analysis techniques in the methods of dynamic malware analysis and dynamic malware analysis tools for the Android operating system.**

***Index Terms*— Android, Malware, Dynamic malware analysis, Botnets, Analysis techniques, Apps, Permissions**

I. Introduction

MOBILE DEVICES have become an integral part of the daily social fabric of our lives and they are the most popular target platforms for attacks. The past year has seen an over 2000% increase in unique malware variant attacks. Recent mobile malware studies have stated that mobile anti-virus tools are likely to catch less than 20% of these attacks. Furthermore, many devices do not have any mobile malware prevention tools installed. With always-connected capabilities, mobile devices have become the ultimate access point to a person's most private and personal information. Once a device has been maliciously compromised, personal data is accessible both locally (on the device) and over the network (in "the cloud"). To top it all, blurring the line between personal and professional is the critical issue relating to "Bring your own device" (BYOD), which impacts both the industry and government sectors.

As the use of mobile devices continues to rise in our personal and professional worlds, there is a growing need to educate future professionals in the topics of mobile security and vulnerability exploitation. Currently in the United States, there are more wireless mobile devices than there are people (331.6M/311.5 at 106%). As the ultimate human computer interface, the mobile device allows almost anyone to do almost anything anywhere. The mobile device transcends space and time through its multitude of communication, transaction, and entertainment tools.

As important and indispensable as these devices have become, security of mobile data and devices is an element that is still largely overlooked. There is a strong sense of urgency to build awareness and protection, and to prevent such mobile device threats. Currently, no academic institution focuses on the lack of security of these personal mobile devices as a threat to private and personal information. Though some programs may add the topic as a week of content into a single cyber security course, this approach is functionally inadequate in addressing the large volume of issues facing the mobile environment.

This paper discusses in detail four labs that have been developed by the author to familiarize the students with the Android platform and dynamic malware analysis techniques. Section II describes the history of the Android operating system. Sections III and IV describe the need for educating students in this field and the need for dynamic malware analysis. Section V describes the first lab, which introduces students to the Android emulator software. The remaining three labs, which introduce the students to dynamic malware analysis, are presented in sections VI, VII and VIII. They present various malware apps that send text messages without the permission of the user, apps used for advertising, and a simple mobile botnet. These labs teach live analysis techniques as well as the principles and methods used by the malware developers and how they trick users into running their malware. Section IX then suggests additional future work in this area and is followed by the conclusion.

II. History of Android Operating System

Google's Android operating system was released to the world in 2008 and is based on a modified Linux kernel, built on the ARM platform. Google claims that each day there are at least 1.3 million activations and the total number of Android devices exceeds 5 million making it the most widely used mobile operating system [1]. Estimates suggest that approximately 75% of the total smartphone devices shipped as of October 2012, were Android devices [2].

C. Szost is with the Rochester Institute of Technology, Rochester, NY-14623, USA (e-mail: crs1471@rit.edu).

K.Sharma is with the Rochester Institute of Technology, Rochester, NY-14623, USA. (e-mail: kxs1203@rit.edu).

Tae Oh is with the Rochester Institute of Technology, Rochester, NY 14623 USA (e-mail: thoics@rit.edu).

Bill Stackpole is with the Rochester Institute of Technology, Rochester, NY 14623 USA (e-mail: wrsics@rit.edu).

Rick Mislan is with the Rochester Institute of Technology, Rochester, NY 14623 USA (e-mail: rpmics@rit.edu).

Android applications are written in Java and its lower level system utilities are written in C language. Its applications can also be written in C++ [3], [4]. However, Java is the preferred language. Google developed a custom "Dalvik virtual machine," which is the replacement for the Java virtual machine on the mobile devices running the Android operating system [5]. This Dalvik virtual machine is customized for use on mobile devices, which typically have limited resources. Thus, an application written in java is compiled into a "dex bytecode" in a file named *classes.dex*.

In an effort to keep the system secure, Android places all user applications into a sandbox when they execute. This allows applications to run independently of one another and not interfere with the resources and memory requirements of other applications. This is done using standard UNIX process separation techniques. As a result, each application is isolated from all others. Each application is assigned a User ID (UID) and a Group ID (GID) when they are installed. There are two ways in which users can install applications into their devices: by downloading from Google's play store or by direct download and installation – with an option to store the app on a memory card. While installing these applications the users are presented with a list of all the permissions that the app requires to be able to execute in the mobile device. The users then have two options: either accept all permissions and allow the app to install or disallow the app and choose to not install it. (No option is provided to the user to choose which permissions to allow or deny – it is an all or nothing selection.)

III. Need for Educating Students in This Field

As mentioned in Section I, malware has been growing tremendously alongside the growth of the Android OS. This calls for a strong need to evaluate the awareness of the users with what malware is and how harmful it can be to their personal and private information. The caveat here, however, is that users are naïve and do not have much of an understanding in this matter as to what apps are genuine and which are malicious. They do not know which permissions are genuinely required by an app and which are malicious and may be used only as a tactic to gain access to private and sensitive information. Most users, rather than reading through and trying to understand *what* permissions are being requested and *whether* it is appropriate for the app to be asking for those permissions, just go ahead and allow the app to access all the information it requested. This is the requirement to permit installation on their devices. Most malware developers make use of this naiveté on the user's part and develop malicious apps, hoping that users not know the difference, will not read through the list of permissions, and will just download and install the app.

This paper presents a concerted effort on the authors' part to educate students regarding the Android platform and to help them gain practical experience with it and its environment; so, when students go out in the world and are faced with such situations, they know what they are dealing with and have a strong conceptual background and basic understanding of what the system does and how it interacts with the apps.

In addition to learning about the Android operating system, the authors also want the students to better understand how apps interact with the underlying operating system, what connections they make, what permissions they request and what API calls they make. By answering those questions, the students can learn and are able to differentiate between genuine apps and malicious apps.

IV. Need for Dynamic Malware Analysis

Dynamic malware analysis is also known as behavior testing or live testing. This allows the user to see what the malware is doing in real-time. It allows the user to see and analyze the behavior of the malware as if it was installed and executing in the actual mobile device. During the dynamic malware analysis, the malware is installed in a virtual mobile environment using an Android emulator. The malware then tries to make the connections as it would if it were installed in a mobile device. However, since this device is virtual and is being monitored by the user, armed with sophisticated tools like tcpdump, users are able to see all aspects of malware behavior such as what processes are running, what network connections are made and what data is transmitted. Based on this behavior, the user can infer conclusions regarding the malware and see exactly what information it steals and how to protect against it.

V. Lab 1 - Android Emulator

The first lab is a brief introduction to Android virtual devices and the steps to set up the testing environment. The labs use a setup of two virtual devices: one innocent device installed by student used as a control, and one device infected with the malware.

The lab provides the student with detailed instructions for the setup for these devices and the environment. It also instructs the student on how to interact with the virtual phone through the adb shell. This gives the user command line access to the underlying Linux operating system on which the Android system runs. This is then used for gathering important information like what processes are running, network statistics, etc. by using the commands like ps and netstat. The ps command will display information about all the processes running on the device and will also include information regarding the name of the user to whom the processes belong to, process ID and the name of the process. This enables the student to see which app is running which processes and if it is normal behavior or if it is malicious. Netstat shows network connections the device has made or attempted to make and the addresses and ports that are in use. In the Android version of these commands are stripped down compared to the true Linux commands, but they can still offer valuable information.

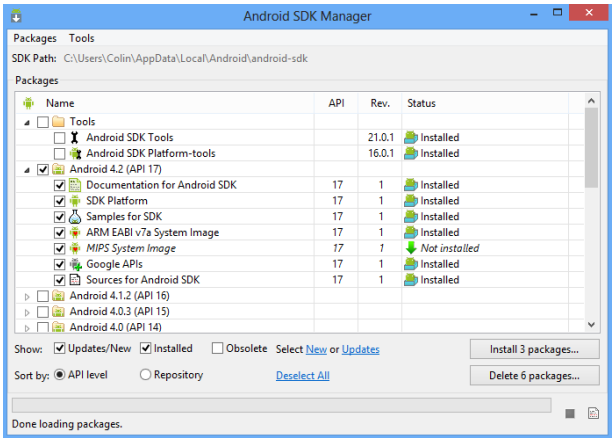


Fig. 1. Android SDK Manager Installation

Another tool introduced was tcpdump. This is the main tool used for analysis and can capture network traffic from a device, provided that the tool has "system" or "root" priviledges. While tcpdump isn't capable of detecting SMS data, it does show all the other connections that the phone makes. It will listen to any network traffic that the device sends or hears and saves it so that the user can look at it afterwards and analyze its content for malicious nature and behavior.

While this lab doesn't introduce any malicious applications, it gives the student a reusable lab environment to work with for the future labs as well as an understanding of how the tools and emulators work. This lab was designed for people who have little familiarity with both Android virtual devices and the Android platform in general.

VI. Lab 2 - SMS Malware

This is the first lab that introduces a malicious app. When picking a sample to use, there were two important characteristics, the first being a simple attack. Students may not be familiar with analyzing network traffic or even the Android platform in general, so it had to be something that wouldn't overwhelm an unfamiliar student. Additionally, a sample that could illustrate how much control a malicious application could have over the phone, and something that would be easily visible and would also be interesting for a student to observe.

A sample that met these criteria was the malware *DogWars – Beta*. This application did two important things. First, it contacted a website. It didn't send or request too much information and was not overly complex. It would give a simple introduction about why DNS would be helpful to malware creators rather than hard coding an IP address, and it would give them an example in what to possibly look for.

The other reason this malware was chosen is because it sends SMS messages without asking user permission. This is both very visible to the student, and more interesting than searching through packet captures. The goals for this lab were to show that a malware can send out mass messages without the user's interaction, knowledge or permission, and the student can see these effects. *DogWars – Beta* also demonstrated an SMS attack, which is one of the types of malware that students may encounter in the wild, and the more different types of malware a student is exposed to, the more they will know what is possible.

The following figure displays the tcp stream that was generated and contains important information regarding the malware network connections.

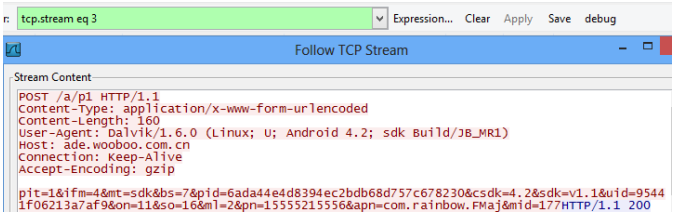


Fig. 2. TCP Stream

At the end of the lab, students are asked a series of questions that point them in the direction of some of the concepts they should be learning through the lab. This malware was rather simple, and the packet captures are easily analyzed.

While not overly complex, this lab gives a good introduction to using the environment, the tools, and the dynamic analysis process. It also demonstrates how easy it is for a malicious program to use a mobile device without the user's knowledge.

VII. Lab 3 - Show Me The Money

The third lab in the series tests a much more complex malware sample. The sample chosen was *LeNa.c,* the Android DFKBootKit variant. This application masquerades as a game. This worked well to show how a trojaned game can function almost completely like the real program while also performing malicious activity.

This application also is more complex than the malware used in lab two. The *LeNa* application sends and requests information in a much less obvious way over the internet to its controller's servers, including information sent in URL strings. The students would have to look for these strings in the packet captures as well as analyze what kind of information is being sent and requested, and determine the purpose of this communication.

The application is used to push advertisements to the user and replaces many of the *in-application* links with links that redirect to malicious websites.

More advanced analysis techniques are also introduced, such as some of Wireshark's advanced traffic analysis features and the student has to look in different places to find what the application is doing.

This sample also shows a different intent for a malware's purpose. While the first sample sent SMS messages as its malicious payload, *LeNa* is made for siphoning information from a device and pushing unwanted ads to the user. This illustrates the various purposes of a malicious program.

VIII. Lab 4 - Spam, It's What's for Spreading

The final lab in this series uses a sample of the Spam Soldier botnet. This is a simple mobile botnet with many useful features that make it ideal for educational purposes. The

main aspect that makes this a good malware sample for an introductory malware analysis course is that command and control functions are not encrypted. This allows students to be able to see and understand the commands being sent to the device without having to completely disassemble the malware.

During the analysis, students will be able to see the command and control messages in their packet captures, and how the application works when the commands are received.

For this lab, a custom command and control server must be set up so that the application accepts commands only from the lab server, and not a real malicious server. This allows students to look at the malware from both the infected device as well as the actual command and control server. This vantage point allows them to see how they interact and what the commands look like.

Another reason this malware sample was chosen was because it shows a different type of malware. Together, all three labs introduce a malicious application, including an SMS attack, a trojan, and a mobile botnet.

IX. Future Work

This paper is a stepping-stone to what the authors wish to be a full-fledged course, complete with many more detailed labs, which increase in complexity with regards to the malware categories and samples. Currently, these labs focus on SMS malware. In future labs, the authors will include malware samples from other malware categories, such as trojans, spyware, bots, and more. Future labs will also focus on helping students identify patterns and distinguishing features of each category of malware. This will enable students to get a deeper understanding of how each category of malware behaves and what they focus on as the end result of malicious behavior.

X. Conclusion

This series of four labs was designed to examine how malicious applications work, and how this malicious activity can be detected and analyzed. They were explicitly designed for those unfamiliar with both Android devices and malware analysis techniques. Currently, this is an incomplete series of dynamic malware analysis techniques labs as there are more advanced aspects of both malware and analysis to be introduced. For this, an advanced course is planned. As an introductory and supplementary set of labs, this initial series provides a broad overview of how common types of malware behave and operate.

The skills and tools learned in this series of labs can be transferred to a more advanced application, and can be used to test if a student is truly interested in this subject. The ideas and techniques taught in this course are also not strictly specific to Android malware, as they can also be used in examining traditional PC malware. Having this basic understanding and familiarity of mobile malware will be helpful to those looking for a deeper understanding of malware analysis topics, such as reverse engineering.

REFERENCES

[1] M. Burns, (2012 September 5). *Eric Schmidt: 'There are now 1.3 million android device activations per day'*[Online]. Available: http://techcrunch.com/2012/09/05/eric-schmidt-there-are-now-1-3-million-android-device-activations-per-day/

[2] IDC, (2012 November 1). *Android marks fourth anniversary since launch with 75% market share in third quarter, according to IDC* [Online]. Available: http://www.idc.com/getdoc.jsp?containerId=prUS23771812

[3] Ohloh, (2012 November 29). *The android open source project* [Online]. Available: http://www.ohloh.net/p/android

[4] Google, (2012 November 1). *Android source code* [Online]. Available: http://source.android.com/source/downloading.html

[5] D. Borstein (2012 November 30). *Presentation of Dalvik VM Internals* [Online]. Available: http://sites.google.com/site/io/dalvik-vm-internals/2008-05-29-Presentation-Of-Dalvik-VM-Internals.pdf?attredirects=0

Discovering Predictive Event Sequences in Criminal Careers

Carl A. Janzen, Amit Deokar and Omar El-Gayar

***Abstract*—In this work, we consider the problem of predicting criminal behavior, and propose a method for discovering predictive patterns in criminal histories. Quantitative criminal career analysis typically involves clustering individuals according to frequency of a particular event type over time, using cluster membership as a basis for comparison. We demonstrate the effectiveness of hazard pattern mining for the discovery of relationships between different types of events that may occur in criminal careers. Hazard pattern mining is an extension of event sequence mining, with the additional restriction that each event in the pattern is the first subsequent event of the specified type. This restriction facilitates application of established time based measures such as those used in survival analysis. We evaluate hazard patterns using a relative risk model and an accelerated failure time model. The results show that hazard patterns can reliably capture unexpected relationships between events of different types.**

***Index Terms*— Predictive analytics, Event sequence mining, Criminal behavior prediction**

I. Introduction

WHEN EVALUATING alternatives for sentencing policy and rehabilitation programs, there is a recurring question of whether or not existing approaches are effective over the long term. One popular approach for quantitative analysis of criminal careers is to cluster offenders according to their offending patterns over time. This approach, called group trajectory modeling, usually results in an offender typology grouping consisting of two to four categories, to which descriptive labels are attached (e.g., short term juvenile, long term chronic offender). Comparisons between these groupings are then made, with attention to correcting for selection bias and exposure time. The concern of selection bias arises when the treatment of interest, such as arrest and incarceration, cannot be randomly assigned. Thus, treatment outcomes may be reflective of factors that influenced assignment to treatment, such as an individual propensity to commit a crime. Corrections for exposure time, or street time, are intended to address the changes in opportunity that may be expected when an individual is incarcerated. This forms the basis for state of the art quantitative studies designed to measure the long-term effectiveness of a particular program of treatment, such as arrest and incarceration. However, approaches based on clustering individuals according to rate of certain events have not yet addressed how an arbitrary number of different types of events throughout the criminal career may affect the possibility of future offenses. Key events in a criminal career include arrest, conviction, sentencing, parole, and discharge. Each of these may be further broken down into sub-types. Since existing quantitative analyses do not facilitate ad-hoc discovery of relationships between events of many different types, unexpected relationships between various different event types remain undiscovered.

Event sequence mining can be used to discover patterns consisting of many different types of events. However, a number of challenges arise with the use of existing measures of interest when used to describe predictive relationships. The most fundamental measure of interest for event sequence patterns is support. This measure indicates the number of pattern occurrences, and is borrowed from association rule mining. For each identified support counting method, at least one of the following limitations applies: (A) length of patterns influences support counting (B) an occurrence may or may not be counted depending on the characteristics of other occurrences of the same pattern (non-independence), and (C) unrelated sub-pattern occurrences unduly inflate support counts of some patterns. These problems do not arise in association rule mining, where there is no dimension of time. A related challenge arises during the analysis of a partial event stream or an event stream with censored observations (observations that are unknown because data is missing or because the observation period ended before the event may have occurred). The challenges raised above need to be addressed in a manner that specifically takes into account the nuances that come with the introduction of the dimension of time.

Insofar as there is an interest in discovering relationships between events of multiple different types, there is a need for a method for the ad-hoc discovery of such relationships. Measures based on occurrences of these patterns should not be unduly affected by pattern length, other occurrences of the same pattern, or unrelated occurrences of sub-patterns. This article includes the following the key contributions from the domain and methodology standpoints. For the criminology domain, we demonstrate that hazard patterns based on occurrences of distinct events can be used to make a statement about expected changes in the probability of certain future events as well as expected changes in time to event. For the

C. A. Janzen is with University of the Fraser Valley, Abbotsford, BC V2V 7B1 Canada (e-mail: carl.janzen@ufv.ca).

A. Deokar and O. El-Gayar are with Dakota State University, Madison SD 57042 USA (e-mail: {amit.deokar, omar.el-gayar}@dsu.edu).

event sequence mining methodology, we address limitations

s: <(p, 1), (b, 2), (b, 3), (b, 4), (c, 5), (c, 6), (p, 7)>

b: burglary, c: conviction, p: parole

Fig. 1. Illustrative event sequence database.

that apply to existing pattern support counting methods, and demonstrate how event hazard patterns address these limitations. We evaluate the usefulness of the event hazard patterns from real data using two time-based models.

The remainder of the paper is organized as follows. In Section II, we introduce the literature in the problem context and highlight unaddressed challenges involved in criminal career analysis and event sequence mining. The gaps identified in this section form the motivation for our design. In Section III, we define the objectives of a solution. These objectives form the guidelines for our evaluation. Section IV includes the design and development of the core algorithms and data structures. In section V, we demonstrate and evaluate the proposed solution, and finally in section VI we conclude with some implications and directions for future research.

II. Problem Identification and Motivation

In this section, we discuss the problem context and the motivation for this work. We provide a review of relevant criminology literature with attention to predictive patterns in criminal careers. We then provide an overview of literature related to event sequence mining and note the needed developments for effective prediction of events in criminal careers.

A. Domain: Criminal Career Analysis

There are a few notable studies that have addressed the challenge of making long-term predictions about criminal history event patterns using a combination of group trajectory modeling and predictive indicators. The group trajectory modeling technique was introduced in [1]. This technique involves clustering offenders into trajectory groups according to offense rate over a period of time. The following three recent studies involve the use of this method.

Group trajectory modeling was used in [4] to cluster individuals trajectory groups, with the goal of predicting membership in chronic (life-long offender) or high rate (frequent offender) groups. Demographic variables as well as the number of early juvenile offenses were considered as candidate predictors of membership in these groups. The sample consisted of all prisoners convicted in the Netherlands in 1977. However, there were no risk factors that were found to be good predictors of trajectory group membership.

The same method was also used to cluster the members of a cohort of adolescent boys in Montreal into groups, in a study examining the effects of adolescent first-time gang joining at the age of 14 [3]. In this case, propensity score matching was used to balance the treatment (joiners) and control (non-joiners). Propensity scores are calculated based on known predictors of group membership, and comparisons between the two groups are made only between individuals with matching propensity scores. The effect of first-time adolescent gang membership at age 14 was associated with a short-term increase in violence, but no other effect was observed.

In [2], group trajectory modeling formed part of a strategy to predict increasing or decreasing offense rate following incarceration, in a cohort of American prisoners released from state prisons in 1994. The researchers included a variable to represent the heterogeneity of the individual offense history in relation to the rest of the trajectory group. Individual offense rate micro-trajectories were estimated for each released prisoner. After a 3 year follow-up period, 40% of the prisoners had an offense rate that was significantly lower than estimated, and 4% of the prisoners had an offense rate that was significantly higher than estimated. However, the analysis did not address arrest hazard beyond the first post-release arrest, or the different types of subsequent events that may occur.

In addition to group trajectory based approaches, where behavior is modeled according to group characteristics, a number of researchers have focused on the predicting the location of the crime. One example of such work is the Blue CRUSH system used by Memphis police [5]. This system is designed to direct enforcement efforts to geographical areas where there is a high likelihood of a crime. Another case is the prediction of hotspots using data from monthly crime reports [6]. Hotspot prediction is based on aggregate figures where the unit of observation is geographical, such as a district. Although the history of a particular area provides useful information for predictive analytics, this approach does not take the histories of individuals into account.

Individual criminal histories are comprised of discrete event occurrences of various types along a timeline, often separated by long periods for which no events of interest occur. Behavior patterns from similar data have been successfully captured using event sequence mining approaches. In [7], event sequence mining was used to effectively capture patterns involving the type and order of activities in a door event log. Event sequence patterns are patterns of events that frequently occur in the same order. These patterns were used to identify five cluster groups within a building, three of which exhibited a strong group membership. However, as far as we know, there is no work applying event sequence mining to the problem of predicting the behavior of individuals.

B. Methodology: Event Sequence Mining

The use of event sequences for predictive modeling poses some unique challenges. Since event sequence mining is an extension of association rule mining, measures of interest commonly used in association rule mining are natural candidates for use in event sequence mining. Two common examples of such measures from association rule mining are support and confidence [8]. However, the use of such measures in event sequence mining is complicated by some fundamental differences between association rule mining and event sequence mining brought about by the dimension of time.

Support is a measure of pattern occurrence frequency, usually expressed as a count. *Confidence* is a measure of association between occurrences of an antecedent pattern and occurrences of a consequent pattern and is calculated as support of consequent / support of antecedent. Confidence that approaches 1.0 shows that when the antecedent is present, the consequent is expected to also be present. Thus, the presence of the antecedent might be used to determine the probability that the consequent is also present.

In the event sequence database in Fig. 1, one individual event sequence is represented. Each event occurrence is associated with the number of months since some point in the past. For our discussion of support counting, we will not consider censored events or relationships between events in one sequence and events in another sequence.

There are two main approaches to event sequence mining: sequence mining, and frequent episode mining. With sequence mining, pattern support is based on the number of input sequences that contain at least one occurrence of a given pattern. With frequent episode mining, it is the prevalence of the pattern without respect to different input sequences that determines support. Since we are looking for predictive relationships within pattern occurrences, we focus on the frequent episode mining approach.

We can apply frequent episode mining to the event database in Fig. 1. Discussion of frequent episode mining with window based support counting can be found in [9]. Using this technique, the event database is subdivided into all possible windows of some specified size ω. Support count is based on the number of fixed size windows that contain at least one pattern occurrence. Using a window size of five months, we see that there are four windows that contain at least one occurrence of *b* and there are two windows that contain *b* followed by *p*. A simple calculation of confidence gives us a 50% confidence that *b* leads to *p* within five months. In this case, the discovered relationship is as follows: 50% of windows of opportunity that contain a burglary event also contain a subsequent parole event. Note that this does not mean that 50% of burglaries are followed by parole. The relationship is with respect to the windows of opportunity.

A number of alternative methods of support counting have been explored in addition to window-based counting. Examples of support counting also include: minimal occurrence based, non-interleaved, non-overlapping, head frequency, total frequency, and distinct occurrence based. Some of these are also commonly combined with an expiry time constraint. For a comprehensive discussion of these variations in support counting, including window-based counting see [10].

TABLE I
SUMMARY OF DOMAIN RELATED GAPS

Approach	Gaps
Group Trajectory Modeling [1-4]	Models expected behavior of individuals over time but does not use multiple event types.
Hotspot based analysis [5, 6]	Relates aggregate counts per geographic region, but does not address individual histories
Event Sequence Mining [7]	Clustering shown to be useful but no examples of event prediction found.

However, the use of these support counting methods for event sequences is hampered by counting that is unduly influenced by pattern length, non-independence of pattern occurrences, and the inclusion of unrelated occurrences. These limitations are detailed in Section IV-B.

Event hazard patterns do not have the above mentioned limitations. These patterns are a specialization of event sequence patterns and are comprised of frequently occurring event sequences, wherein each event in the event sequence is the first subsequent occurrence of that event type [11].

Table II contains event hazard patterns based on the contrived event sequence in Fig. 1. The first burglary charge following parole release leads to a higher proportion of subsequent burglary charges when compared to the remaining cases. Note that there are three opportunities for a burglary charge to be repeated. However, after accounting for the one burglary charge that immediately follows a parole release, only two remain. Thus we have a 100% confidence that parole followed by burglary will lead to more burglary, but we only have 50% confidence that subsequent burglaries will do the same (1/1 instead of 1/2 for the remaining burglary charges). Naturally, this does not give us an indicator of generalizability nor does it account for censored observations. We will address each of these in Section IV.

Since hazard patterns incorporate information about the interval that precedes the first occurrence of each subsequent event, we expect them to be well suited for time-based analysis. Time-based models are well suited for addressing ordering of events, and include methods to deal with censored events.

Two complementary time-based options are relative risk ratio (RR) and accelerated failure time (AFT) models. RR is an indicator of treatment impact that relates the number of failures in the treatment group to the number of failures in the control group [12], [13]. In contrast, AFT models the relationship between the expected time before failure in the treatment group, relative to the same in the control group [14].

There is a substantial body of literature in the field of developmental criminology involving criminal career trajectory analysis, but there is still a need for a method to discover interesting relationships between the many different types of events in a criminal history. Existing approaches to

quantitative criminal career analysis focus primarily on offense rate predictions.

Due to the limitations outlined above, measures adapted from association rule mining, such as support and confidence may not adequately capture predictive relationships between time-ordered events. Since event hazard patterns do not have these limitations, we expect them to be well suited for integration with time-based models such as RR and AFT.

III. Objectives of the Proposed Pattern Discovery System

Existing quantitative approaches to criminal career analysis are not well suited to the ad hoc discovery of relationships between different event types. Event sequence mining is a potential method for the discovery of such relationships.

Event hazard patterns have not yet been used with time based measures of interest. In this work we empirically evaluate the predictive ability of event hazard patterns selected using time based models.

We propose a software instantiation to address the challenge of ad-hoc discovery of predictive event sequences in criminal careers. The main objectives of the proposed pattern discovery system are:

1) Discovery of frequent event sequences in a database of criminal career events
2) Selection of accurate predictive patterns

In this work, we make two key contributions:

3) A domain contribution: facilitating the ad-hoc discovery of relationships between various different event types in a criminal history
4) A methodology contribution: introduce the use of time-based models with event hazard patterns

IV. Design and Development

The proposed pattern discovery system builds on an event hazard pattern discovery algorithm. A crime analytics system that will utilize this pattern discovery system is currently under development. In this work we adapt the pattern discovery system for use with time-based measures of interest.

The pattern discovery algorithm is designed to facilitate discovery of event patterns in an event history database expected to contain frequent event sequence patterns separated by both short and long time intervals during which each subsequent event does not yet occur. Such patterns are a specialization of event sequence patterns and are referred to here as event hazard patterns. Given that we expect time based event occurrences that are independent from each other to occur at intervals that follow an exponential distribution, we apply hazard constraints to approximate intervals of exponentially increasing size. This strategy is described as heterogeneous constraints in [11].

A. Definitions

Except where specifically noted, the following are definitions of terms commonly used in event sequence mining. For further details on these terms see [9] and [10].

TABLE II
Event Hazard Patterns

Pattern	Opportunities	Support
b	-	-
$b \rightarrow b$	3	2
$p \rightarrow b \rightarrow b$	1	1

Event Type: An event-type refers to a class of discretely identifiable events with common characteristics.

For example, when an individual is arrested charged with a burglary offense, an event type of burglary arrest charge occurs. Additionally, it can be said that a more general event type of arrest charge, or property crime related arrest charge occurs at the same time. An event type is alternatively referred to as an event.

Event Occurrence: The occurrence of an event is denoted (e, t), where e represents the event type and t represents the time of the event occurrence. The unit of discretization for t, such as second, minute, hour, day, etc. is an important consideration when selecting constraints that must be satisfied by t. For example, $(c, 5)$ is the occurrence of event (or event type) c at time 5.

Event Sequence: An event sequence of length n is denoted $<(e_1, t_1),(e_2, t_2),...(e_n, t_n)>$ where e_i represents the type of the i^{th} event, t_i represents the time of the i^{th} event, and $t_{i-1} < t_i$. An event sequence is a time oriented arrangement of event occurrences. For example, $<(b,4),(c,5)>$ is an event sequence. In this work we address only serial event sequences.

Event Sequence Pattern: A frequently occurring event sequence, as defined by a minimum support threshold. An event sequence pattern can be denoted as $<(e_1, T_1), (e_2, T_2),... (e_n, T_n)>$, where e_i represents the type of the i^{th} event occurrence, and T_i represents the collection of all occurrences of the i^{th} event type. Each occurrence represented in T_i with $i>1$ corresponds to an antecedent occurrence represented in T_{i-1}. Alternatively, an event sequence pattern can be summarized in a more compact and intuitive form, as a sequence of events: $b \rightarrow c \rightarrow p$.

In the context of sequential pattern or sequence mining, an event sequence is frequent when it occurs in many input sequences. In the context of frequent episode mining, an event sequence pattern is frequent when there are many occurrences of the pattern. In this work, we consider event sequence mining in the context of frequent episodes.

Gap Constraint: The requirement that except for the initial event occurrence, for any event occurrence (e_i, t_i) in an event sequence, there exists at least one event occurrence (e_{i-1},t_{i-1}) where $mingap \leq (t_i-t_{i-1}) \leq maxgap$. For example, two events in an event sequence satisfy a minimum gap constraint if they are separated by at least *mingap* and they satisfy a maximum gap constraint if they are separated by at most *maxgap*.

The selection of appropriate *mingap* and *maxgap* are domain specific. Gaps are chosen by a human operator to reduce the number of irrelevant patterns that are discovered.

Hazard Constraint: The requirement that except for the initial event occurrence, for any event occurrence (e_i,t_i) in an

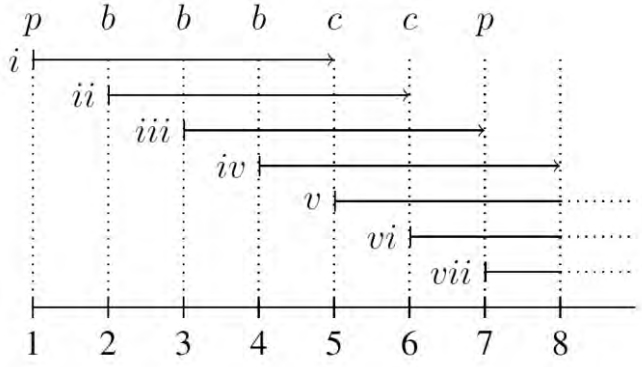


Fig. 2. Windowing and unrelated occurrences.

event sequence there exists no event occurrence (e_r, t_r) where $e_r = e_i$ and $t_{i-1} < t_r < t_i$. Furthermore, each antecedent occurrence (e_{i-1}, t_{i-1}) is a unique occurrence. In other words, each successive event occurrence in an event sequence is the first successive occurrence of the specified type, relative to occurrences of the specified antecedent event type in the same event sequence. Hazard constraint is a new term first introduced in [11].

As with *mingap* and *maxgap*, a hazard constraint can be expressed as a *minhaz* and *maxhaz* for similar effect. A hazard constraint *h((min,max])* specifies an interval during which the subsequent event may first occur, such that for all event occurrences of a given event hazard pattern, $minhaz < (t_i - t_{i-1}) \leq maxhaz$. In other words, the first occurrence of each subsequent event type, relative to the antecedent occurrence, takes place after *minhaz* and may occur as late as *maxhaz*.

Hazard Pattern / Event Hazard Pattern: An event sequence pattern wherein each subsequent event occurrences the first subsequent occurrence of that particular event type. A specific hazard constraint of *minhaz* and/or *maxhaz* may also be specified, to select only those cases where the subsequent event first occurs after *minhaz* but no later than *maxhaz*. A given event hazard pattern $a \rightarrow b$ can be expressed with a hazard constraint as *a (minhaz, maxhaz] b*. Hazard Patterns were recently introduced in [11]

Relative Support: The number of unique antecedent event occurrences that are followed by a subsequent event type in an event hazard pattern. For example, in Fig. 2, $c \rightarrow p$ occurs twice, but $c \rightarrow c$ occurs only once. Relative support was proposed for event hazard patterns in [11].

Relative Risk: The ratio of the risk within a treatment group over the risk of the control group. It is used to measure the cumulative treatment effect at the end of a period of time. For a discussion of practical application of relative risk ratios, see [13].

B. Justification for Relative Support

There are ten methods of counting support described in [10]. However when using these support counts to describe sequential relationships, a number of challenges arise. Relationships between events in an event sequence are not representative when support count is affected by (a) length of patterns, (b) non-independence between pattern occurrences, and (c) side effects of unrelated pattern occurrences on support count. Further, in the case of incomplete or censored observations, we need to draw on time based analysis methods. We discuss each of these challenges in detail below.

Length of patterns: When the number of pattern occurrences is directly dependent on the length of the pattern, it is difficult to use differences in support count to construct sequential relationships between shorter and longer extensions of those patterns. We generally expect longer patterns to occur less frequently, but window-based counting methods further penalize the support count of longer sequences. One example of this phenomenon is with event sequences that are constrained by a window of opportunity or expiry time. For instance, the event sequence <*(b,3),(c,4)*> in Fig. 2 appears in four windows of opportunity of size five whereas the event sequence <*(b,3),(c,4),(c,5)*> occurs in only three windows of opportunity.

Independence: An alternative to window based counting is occurrence-based counting. Examples of these are minimal occurrence based, non-interleaved patterns, non-overlapping patterns, and distinct occurrence based counting. Each of these suffers from a lack of independence between pattern occurrences. This is because in each of these cases, some pattern occurrences are not counted based on the position and ordering of events in other occurrences of the same pattern. Examples of such missed counts are detailed in [15]. Violation of the independence assumption makes it more difficult to describe relationships between patterns using statistical methods.

Unrelated Occurrences: A solution to the problem of non-independent occurrences is to use head or total frequency. With head frequency, the number of pattern occurrences is based on the number of windows of a specified size that start with the head (first event) of the pattern [16]. However, the same challenges described for other window-based counting methods still apply to head frequency. In addition, head frequency has the undesirable side effect of over-representing the number of occurrences of patterns with a frequent head.

For example, in Fig. 2 the pattern $b \rightarrow c \rightarrow c$ has a head support of three (windows *ii, iii*, and *iv*). However, the support count is unduly inflated by the relationship represented in the initial $b \rightarrow c$ sub-pattern. Total frequency is a partial remedy to this problem whereby the support is equal to the lowest head frequency of any sub-pattern [17].

However, this measure can still be unduly inflated by unrelated occurrences of sub-patterns. For instance, in Fig. 2 *c* and *p* both have a head frequency of two, so the total support of $c \rightarrow p$ is two even though the support count is affected by an occurrence of *p* that is unrelated. Thus, we cannot use this measure to describe relationships between antecedent patterns and their subsequent extensions.

Censoring: In Fig. 2 windows *v, vi* and *vii* are all censored. If the patterns are not sufficiently short relative to the event database, this missing data may adversely affect the interpretation of support counts. Missing or incomplete event data, if not accounted for, can be misleading. In the real database of criminal histories used in our analysis, the

Require: $\forall(event, ordinal) \in D : event \in M$
Require: Constraints G
Ensure: $\forall Pat \in Freq : |Pat| \geq sup$
1: **function** GROW(Pat, $Ords$)
2: **for all** $event \in M$ **do**
3: **for all** $constrt \in G$ **do**
4: **if** $|Ords_{constrt,event}| \geq sup$ **then**
5: $NewPat \leftarrow (Pat, constrt, event)$
6: append $NewPat$ to $Freq$
7: $Nextords \leftarrow$ NEXT($Ords_{constrt,event}$)
8: GROW($NewPat$, $Nextords$)
9: **end if**
10: **end for**
11: **end for**
12: **end function**

Fig. 3. Pattern discovery.

database is considered to be complete, such that events past the end of each criminal history are believed to be negligible. However, the matter of censoring in event sequence patterns is problematic for cases where pattern length is not negligible relative to the length of input sequences. Event hazard patterns do not specifically address event censoring, but these can be addressed with time-based models. For a related discussion of censored events in event sequence mining with variable window sizes, see [18].

Some additional limitations of existing support count methods are also discussed in [15] along with a complex proposed support metric. The proposed support metric relies on non-redundant occurrences (occurrences with no events in common). This non-redundancy requirement introduces dependencies between different occurrences of the same pattern.

With consideration to the limitations outlined above, Hazard patterns are counted by relative support. Given a number of distinct opportunities, support is the number of those distinct opportunities or distinct antecedent events that are followed by the subsequent event. The number of opportunities is less than or equal to the support of the antecedent pattern. Note also that, unlike the event sequence mining approaches described above, each subsequent event in an event hazard pattern is the first such subsequent event, and that it is not necessarily distinct (it may participate in more than one pattern occurrence).

For instance, in Fig. 2 there are three occurrences of *b*. All three of them are followed by *c*, so support for $b \rightarrow c$ is three. However, since these three antecedents all converge on the same *c* occurrence at position four, there is only a single distinct opportunity to extend $b \rightarrow c$ to the subsequent *c* or *p* events. The support of $b \rightarrow c$ is three (out of three distinct opportunities), and the support of $b \rightarrow c \rightarrow p$ is one (out of one distinct opportunity). Using this approach, confidence is the proportion of successes given a number of distinct opportunities, meaning the confidence of $b \rightarrow c$ is 1.0 and the confidence of $b \rightarrow c \rightarrow p$ is also 1.0.

In this way, hazard patterns address all of the challenges discussed above except censoring. Length of patterns does not unduly affect support count, pattern occurrence counting treats

Require: $[R_{event,ordinal}]_{(m \times n)}, [I_{event,ordinal}]_{(m \times n)}$
Require: alphabet of all event types M
1: **function** NEXT($Antecedents$)
2: $Antecedents \leftarrow$ UNIQUE($Antecedents$)
3: **for all** $event \in M$ **do**
4: **for all** $ord \in Antecedents$ **do**
5: $c \leftarrow R_{event,ord}$ ▷ get constraint
6: $nextOrd \leftarrow I_{event,ord}$
7: append $nextOrd$ to $Subsequents_{event,c}$
8: **end for**
9: **end for**
10: return $Subsequents$
11: **end function**

Fig. 4. Get next ordinals by event, constraint.

each pattern occurrence independently, and events that are unrelated to the relationship do not affect relative support counts. However, although the confidence measure provides an indicator of the proportion of success, it does not provide information about the generalizability of the pattern. To this end, and to account for censored observations, we draw on time-based models in our analysis.

C. Algorithm Design

It is expected that some frequent event sequences will include events that occur close together and others that occur far apart. One way to capture such patterns is to use a windowing strategy, first described in [19], to create item sets, alternatively presented as a partial order or as parallel episodes [20]. However, this approach may discard potentially valuable information, and relies on the analyst to specify optimal windowing and gap constraints.

Windowing and gap constraints capture all events that fall within the constraint boundaries, but do not differentiate between them and do not capture the non-occurrence of an event. Instead, hazard constraints specify a period during which an event does *not* occur, followed by a period during which the first occurrence an event *does* take place.

The proposed algorithm iteratively applies hazard constraints of exponentially increasing sizes, similar to the use of multiple periodic constraints in [21]. The proposed implementation uses progressively larger intervals to represent the number of months before the first occurrence of the subsequent event, such as re-arrest following discharge or parole release.

The GROW function shown in Fig. 3 uses pairs of event and ordinal values to represent a database of known event occurrences. Ordinals are translated to offsets at $O(1)$ cost as needed for constraint calculations. Input ordinals are supplied in a matrix indexed by *event,constraint* where each $M_{event,constraint}$ represents the antecedent ordinals for the current pattern growth step. In Line 4, those antecedents with cardinality that is high enough to meet a specified support threshold are added to the frequent pattern database in line 4, and are passed to the NEXT function, where a new matrix of candidate event occurrences is created, and passed to the subsequent recursive GROW attempt on line 8.

Fig. 4 contains the NEXT function, which takes as input a

collection of antecedent ordinals, grouped by event, and produces the Ordinal matrix *NextOrds* needed in line 7 of Fig. 3. This function uses two indexes: $R_{event,ordinal}$ and $I_{event,ordinal}$. See Fig. 5(c) and 5(d) for the *R* and *I* indexes corresponding to event sequence τ shown in Fig. 5(a). Ordinals in *I* have corresponding offsets in Fig. 5b and constraint identifiers in *R* have corresponding constraint intervals in Fig. 5(e). *R* and *I* are matrices of dimension *(m* × *n)* where *m* is the alphabet of all possible events, and *n* is the number of distinct offsets. Multiple events may occur at the same offset. *I* contains the ordinal of the subsequent occurrence of a given event type. The value stored at the intersection specified by an ordinal and an event type corresponds to the ordinal of the first subsequent occurrence of that event type. *R* contains the constraint that is satisfied at a given event offset (represented as an ordinal), relative to its immediate antecedent event.

On line 5 of the NEXT function pseudo-code in Fig. 4, for each antecedent event occurrence, the constraint $R_{event,ordinal}$ that is satisfied for each potential subsequent event is retrieved. Given the half-open interval topology used to describe the different constraints, each subsequent event can satisfy one constraint. In line 6 the subsequent ordinals are retrieved from *I* and then grouped according to their matching constraints in line 7. The creation of *R* and *I* are not described here, but are straightforward. Their purpose is to pre-compute comparisons and look-ups that are frequently repeated during candidate generation. Simply put, the index serves to reduce the number of calculations required during candidate generation at the cost of increasing memory usage up front. Optimization strategies to take advantage of redundancies in *R* and *I* are currently being evaluated.

V. EVALUATION

The goals in this undertaking involve both a domain and a methodology contribution. For the problem domain, this work provides a method for discovering predictive sequences of events. The methodology contribution is the use of a time-based measure of interest to demonstrate the generalizability of discovered relationships.

A. Data Preparation

The pattern discovery system was used to discover patterns in a data set of complete criminal histories. The histories were collected from part of a non-random sample of offenders who entered the California Youth Authority's Deuel Vocational Institute in 1964 and 1965. The event database contains of 54,175 arrest records and associated dispositions, parole, and discharge events for 3,652 individuals from the time of first arrest through 1983. Dates were discretized to the nearest 15th day of the month [22].

For this analysis, the individual histories in the dataset were randomly assigned to either the training set or the testing set. Each arrest event was associated with up to five arrest charges. Additionally, the nature of the disposition and judgment date was also recorded for each arrest event, as were parole and discharge events. Arrest charges were encoded both as the specific arrest charge as well as a general arrest event. Disposition events were similarly encoded, with the additional adaptation that arrest dates were used for disposition events. Note that due to the discretization of the data, the relationship between an arrest and a conviction for that same arrest is not represented. All dispositions (including convictions) were recoded to the arrest charge date. Any patterns showing both arrests and convictions have nothing to do with conviction rates.

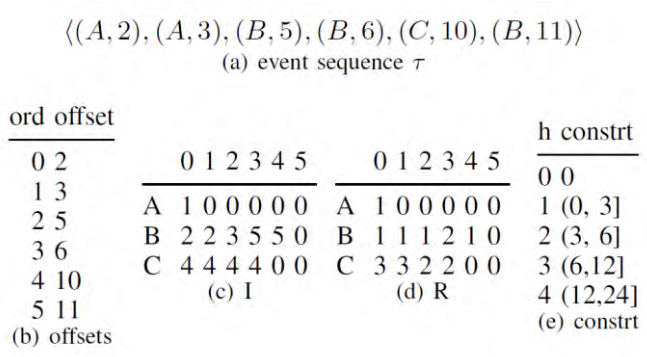


Fig. 5. Ordinal, constraint, and offset indexes

TABLE III
OCCURRENCE FREQUENCIES
FOR PATTERN P AND BASELINE B

	Occurrences	Non-occurrences	Total
P	(a) 775	(b) 914	1689
B	(c) 1146	(d) 2648	3794
Total	1921	3562	(n) 5483

Hazard pattern mining was performed on the training data with multiple different hazard constraints per pattern at increments of 0, 3, 6, 9, 12, 24, 48, 96, 192, and 384 months to generate hazard constraints of (0,3], (3,6], (6,12], and so forth (see heterogeneous constraints described in [11]). Only patterns with a support count of at least 500 were mined. This process yielded 44 frequent events and 1085 hazard patterns (single events are not considered hazard patterns). For each hazard pattern the number of opportunities, the support count (number of opportunities that were successful) and the number of unique subsequent events were recorded. The same patterns were also mined from the test set.

Of the 1085 hazard patterns, 305 patterns involved an event sequence of three or more events. Each of these was compared against the equivalent patterns from the test set and against the baseline or control pattern with the first antecedent and constraint removed. For example: $a \rightarrow p \rightarrow a$ from the training set would be compared with the same pattern in the test set, as well as against a minimally differentiated baseline of $p \rightarrow a$ from the training set. There should be no statistically significant difference between the training set and the test set. Further, there should be agreement between the test set and the training set about the expectations implied by the discovered patterns.

B. Relative Risk

One measure considered for this evaluation was Relative Risk Ratio (RR) [12], [13]. Since the RR measure is applied only at the end of a follow-up period, the measure is inherently sensitive to the choice of follow-up. Further, since relative risk does not account for left truncated data, we use this measure when there is no minimum hazard constraint. For this reason we evaluate only those patterns that end with a hazard constraint of greater than zero and up to three months, expressed as (0,3].

Relative Risk (RR) is defined as follows:

$$\text{RR} = \frac{\text{Estimated risk in the exposed group}}{\text{Estimated risk in the unexposed group}}$$

For our analysis, we consider the set difference between the baseline pattern and the pattern of interest to be the unexposed group. Consider the following two patterns:

P: "Arrest" (0,3] "Continue probation"(0,3] "Arrest"
S: "Continue probation" (0,3] "Arrest"

To construct a baseline, we calculate the measures for *B = (S-P)*. Since *S* includes all pattern occurrences that participate in *P*, we subtract *P* from *S* to create a baseline *B* to compare against. In other words, *B* contains all occurrences in *S* that do not also occur in *P* (see Table III). Thus, we can measure the relative risk of "Arrest" within three months associated with a disposition of "Continue probation" occurring within three months of an arrest compared to those cases where it did not occur within three months of a preceding arrest.

Referring to Table III, we can calculate

$$RR = \frac{(a/(a+b))}{(c/(c+d))} = \frac{775/1689}{1146/3794} \approx 1.52$$

We see that the risk of re-arrest within three months in the final stage of pattern *P* (0.46) is 1.52 times the risk of re-arrest within three months in pattern *B* (0.30) for an absolute risk difference of 0.16%.

It is important to note at this stage that the pattern does not provide enough information to state that the initial "Arrest" and delay before the "Continue probation" disposition event were the key predictors. Although this may seem to be a perfectly intuitive conclusion, it would overlook the unrelated occurrence problem described in Section IV-A. To put it another way, even though all occurrences of *P* contain an occurrence of *B*, some occurrences of *B* may have occurred without a coinciding occurrence of *P*. There may be such a relationship, but it is not represented by these counts.

Instead, what is represented is the relationship between the entire antecedent pattern and the last constraint plus event combination. We can compare the effect represented by *P* to the effect in the shorter pattern *B* to determine whether the additional information provided by the longer pattern may be of use. To this end, we estimate the confidence interval for the RR calculated above. The standard error is symmetrical about the logarithm of RR as follows:

$$SE(\ln RR) = \sqrt{\frac{1}{a} - \frac{1}{a+b} + \frac{1}{c} + \frac{1}{c+d}}$$

A 95% confidence interval is then estimated by taking the antilog:

$$e^{\ln(RR) \pm SE(\ln RR) * 1.96}$$

Similarly, a Z score can be estimated as follows:

$$Z = \frac{\ln RR}{SE\ (\ln RR)}$$

This makes it possible to perform a test to see if the difference in relative risk ratios is due to chance. See [13] for an example. For patterns *P* and *B*, the resulting z-score is 11.56, supporting a rejection of the null hypothesis that the difference between *P* and *B* is due to chance.

Further, we can show the robustness of the pattern by comparing the risk ratio of *P* in the training set with the risk ratio of *P* in the test set. The risk ratio for *P* in the test set was 748 / 1693. Using the same process outlined above, we estimate a relative risk ratio of 1.02, showing that the two risk ratios are almost the same (RR=1.00 would indicate no difference at all). We then calculate a z-score of 0.48 showing that we are unable to reject the null hypothesis that the differences between *P* from the training set and *P* from the test set are due to chance.

For this portion of the analysis, we concern ourselves only with the patterns ending with a constraint of (0,3] due to the limitations of the relative risk measure discussed above. Of the 305 patterns that could be paired with a baseline, 66 patterns have an ending constraint of (0,3]. In other words, for this evaluation, we consider only patterns with at least three events, and with the constraint between the last two events being greater than zero and up to three months.

Based on a 95% confidence interval, 47 of the 66 selected patterns were shown to represent a statistically significant difference in risk between discovered patterns and their corresponding baselines. Further, 2 of the 66 selected patterns were found to have statistically significant differences between the risk ratios discovered in the training set when compared to the risk ratios discovered in the test set and there were no instances where patterns in the training set indicated a significant increase while the corresponding patterns in the training set indicated a significant decrease and vice versa. In other words, the training and test set did not contradict each other.

The risk difference between the selected 66 patterns and their corresponding baselines ranged from -0.07 to 0.16. The range of values for risk (support/opportunities) for the discovered patterns was 0.08 to 0.50.

C. Accelerated Failure Time

Whereas RR is a measure indicating the difference in the number of people affected by treatment when compared to a control group, accelerated failure time (AFT) models show the difference in expected time to event for the two groups.

AFT models are described in detail in [14]. For this analysis we used the Survival package for the R statistical software platform [23]. However, since the data being analyzed is discretized to the nearest month, and AFT expects continuous values, the events were analyzed as interval censored. In other words, the event occurred within a one month interval, but the exact time is unknown. It may be fair to state that all measurements are discretized to some degree, but in this case, a conservative approach was taken. An AFT model was constructed without regard to age, and the results were evaluated in the same manner as with RR above. Although we can expect improved results by taking into account time-varying coefficients such as age, and non-varying predictors such as demographics, our focus for this work is on the efficacy of the discovered patterns themselves.

The same 66 patterns described above were each individually used to fit an AFT model, using a logistic distribution. Other distributions that were tested were Weibull, lognormal, exponential, and Gaussian. In all cases, patterns in the training set were not contradicted by patterns in the test set. The logistic distribution was selected because it produced the most consistent results across training and test sets. Of the 66 patterns, based on a 95% confidence interval, 39 patterns showed a significant decrease in time to next event, and three patterns showed a significant increase. In 24 cases, there was no significant difference between pattern and baseline. AFT models were also fitted to compare the patterns found in the training data with the patterns found in the test data. Only one of the 66 patterns was found to have differences that cannot be attributed to chance. Furthermore, as we found with RR, there were no cases where the models fitted to the training and the testing data presented directly contradictory results.

VI. Conclusion

In this work we explored the use of time based measures for rule selection to address the problem of predicting criminal behavior. Although there has been some limited use of time based measures in event sequence mining, some characteristics of existing methods of event sequence counting make it difficult to accurately discover predictive relationships. These characteristics are (A) support count methods that unduly penalize longer patterns, (B) support count methods that involve dependencies between occurrences of the same pattern (an occurrence may or may not be counted depending on characteristics of other occurrences of the same pattern), and (C) support count methods that include unrelated event occurrences (sub-pattern or event occurrences that do not participate in a relationship with the super-pattern). Hazard patterns do not suffer from these limitations. We demonstrate the utility of hazard patterns for discovering sequential relationships between diverse event types. Patterns were selected and evaluated using two time-based methods: relative risk ratio, and accelerated failure time models.

We note a number of important limitations. First, the relative risk ratio is not suitable for multiple follow-up periods. Patterns with follow-up periods other than (0,3] were excluded from the relative risk analysis. Further, since the event data is discretized to the nearest 15th day of the month, some short term patterns, such as an arrest event leading to a disposition event in less than one month were excluded during data preparation. Additionally, relative risk is sensitive to choice of follow-up period. For instance, the outcomes may have been different if we had selected (0,6] or (0,12] as the follow-up period.

Further, although care was taken to ensure that opportunity and pattern occurrence counts did not violate the independence assumption, some questions affecting generalizability remain. For instance, since the individuals were not randomly assigned to patterns and their respective baselines, a concern over selection bias is justified. The pattern occurrences were not matched or balanced to correct for selection bias. However, given the stability of the patterns between the training and the test set, we did not find evidence of a significant selection bias effect. However, in this work we evaluate only the directionality of effect. We may encounter evidence of bias upon examination of predicted effect size.

A number of directions for further work have been noted. We plan to explore the use of hazard ratio and survival curves to describe the effect over time that is represented by a particular pattern. Other available covariates, particularly age, may improve the accuracy of event hazard patterns. Further, the existing pattern database was mined at a relatively high support threshold. It remains to be seen how robust these patterns are, and particularly how useful time based measures of interest will be when the minimum support threshold is lowered. Another natural extension of this work is the use of sensitivity analysis to address concerns of selection bias. Finally, the use of the techniques described in this work can reasonably be extended to other domains where there are many different types of antecedent events, and where time before the first subsequent event is important.

References

[1] D. S. Nagin and K. C. Land, "Age, criminal careers, and population heterogeneity: specification and estimation of a nonparametric, mixed Poisson model," *Criminology,* vol. 31, pp. 327-362, 1993.

[2] A. S. Bhati and A. R. Piquero, "Estimating the Impact of Incarceration on Subsequent Offending Trajectories: Deterrent, Criminogenic, or Null Effect?," *The Journal of Criminal Law and Criminology,* vol. 98, pp. 207-253, 2007.

[3] A. Haviland, D. S. Nagin, and P. R. Rosenbaum, "Combining propensity score matching and group-based trajectory analysis in an observational study.," *Psychological methods,* vol. 12, pp. 247-67, September 2007.

[4] B. E. Bersani, P. Nieuwbeerta, and J. H. Laub, "Predicting Trajectories of Offending over the Life Course: Findings from a Dutch Conviction Cohort," *Journal of Research in Crime and Delinquency,* vol. 46, pp. 468-494, September 2009.

[5] J. Vlahos, "The Department of Pre-Crime," *Scientific American,* vol. 306, pp. 62-67, December 2011.

[6] C.-H. Yu, M. W. Ward, M. Morabito, and W. Ding, "Crime Forecasting Using Data Mining Techniques," *Proceedings of the 11th International Conference on Data Mining Workshops,* pp. 779-786, December 2011.

[7] T. Abraham, "Event sequence mining to develop profiles for computer forensic investigation purposes," in *Proceedings of the 2006 Australasian workshops on Grid computing and e-research-Volume 54*, ed: Australian Computer Society, Inc., 2006, pp. 145-153.
[8] M. Spiliopoulou, "Managing interesting rules in sequence mining," *Principles of Data Mining and Knowledge Discovery,* pp. 554-560, 1999.
[9] H. Mannila and H. Toivonen, "Discovering frequent episodes in sequences extended abstract," in *1st Conference on Knowledge*, ed, 1995.
[10] A. Achar, S. Laxman, and P. S. Sastry, "A unified view of the apriori-based algorithms for frequent episode discovery," *Knowledge and Information Systems,* May 2011.
[11] C. A. Janzen, A. V. Deokar, and O. F. El-Gayar, "Non-parametric discovery of event sequence patterns in criminal behavior," in *Proceedings of the 46th Annual Hawaii International Conference on Systems Science (HICSS-46 '13) Symposium on Credibility Assessment and Information Quality in Government and Business*, ed. Maui, HI: IEEE Computer Society, 2013.
[12] L.-A. McNutt, "Estimating the Relative Risk in Cohort Studies and Clinical Trials of Common Outcomes," *American Journal of Epidemiology,* vol. 157, pp. 940-943, 2003.
[13] V. Bewick, L. Cheek, and J. Ball, "Statistics review 11: assessing risk.," *Critical care (London, England),* vol. 8, pp. 287-91, 2004.
[14] R. Cook and J. Lawless, "The statistical analysis of recurrent events," 2007.
[15] M. Gan and H. Dai, "Reliable Knowledge Discovery," 2012.
[16] M. Gan and H. Dai, "Fast mining of non-derivable episode rules in complex sequences," *Modeling Decision for Artificial Intelligence,* vol. 1, pp. 67-78, 2011.
[17] K. lwanuma and H. Nabeshima, "On anti-monotone frequency measures for extracting sequential patterns from a single very-long data sequence," in *IEEE Conference on Cybernetics and Intelligent Systems, 2004.* vol. 1, ed: IEEE, 2004, pp. 213-217.
[18] N. Müller, M. Studer, G. Ritschard, and A. Gabadinho, "Extraction de règles d'association séquentielle à l'aide de modèles semi-paramétriques à risques proportionnels," *mephisto.unige.ch,* 2008.
[19] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proceedings of the Eleventh International Conference on Data Engineering*, ed: IEEE Computer Society, 1995, pp. 3-14.
[20] H. Mannila, H. Toivonen, and A. I. Verkamo, "Discovery of Frequent Episodes in Event Sequences," *Data Mining and Knowledge Discovery,* vol. 289, pp. 259-289, 1997.
[21] J. Han and G. Dong, "Efficient mining of partial periodic patterns in time series database," *Data Engineering, 1999. Proceedings.,,* pp. 106-115, 1999.
[22] E. Wenk. (2006). *Criminal Careers, Criminal Violence, and Substance Abuse in California, 1963-1983*. Available: 10.3886/ICPSR09964.v1
[23] T. M. Therneau, "A Package for Survival Analysis in S," ed, 2012.

A Conceptual Investigation: Towards an Integrative Perspective of Risks in Information Systems Development & Usage

Jim Samuel

***Abstract*—A review of existing literature in academia and in profession, along with real cases, reveals a fragmented approach to risk identification and management in information systems development and usage (ISDU). Such a disjointed approach to risk management fails to consider critical threat components and under evaluates the maximum potential risks involved in a situation. The present study argues that ISDU Risks need to be expansively identified and perceived in an integrative manner. The views generally exercised by IS researchers, within the limited attention provided to risk, are segmented and microscopic with no defined risk ecosystem to place them in. Also, practitioner groups have been individualistically producing IS artifacts for risk mitigation with a primary purpose of creating significant ROI, adding to fragmented perspectives on ISDU risks. We observe high failure rates for IS projects, in spite of the claims of the application of highly advanced risk management models. In any domain, the presence of an abnormally high rate of failure would imply an absence of successful risk management and imply that not all significant risks have been accounted for. The present study identifies various cross-domain risk measures and risk constructs with macro-level relevance to the ISDU ecosystem. Based on literature review and observational reflections, a taxonomy for the classification of types of risks is presented. The present paper is an attempt at expanding the portfolio of risk concepts associated with ISDU and posits an early stage high-level integrative risk perception framework that will represent various cross-domain measures and dimensions of risk in an integrative manner. This theoretical contribution and its continued development is expected to initiate additional scholarly work on integrative perspectives on risks and new dimensions of risks associated with IS, open up a new stream of risk-related research in IS and lead to the development of enhanced risk management models.**

***Index Terms*— Risk, Information systems, Volatility, Macro, Mesa, Micro, Uncertainty, VaR, Integrative, Variance, Constraints, Control, temporality.**

I. Introduction

WITH THE continued rapid advancement of computing technologies, we see a significant rate of progress in the efficiencies and effectiveness driven by advances in information systems. These advances have created a complex ecosystem of technologies and this dynamic is aptly captured by World Economic Forum's Executive Chairman Klaus Schwab[1] *"... we need a new model to master the trend of technology. The velocity of technological change, for which we are not really prepared, will accelerate in an exponential manner, having significant implications... ...What is particularly striking, for me as an engineer I may add, is the character-changing nature of technological change..."*. Attention is also being drawn towards an uncomfortably challenging observation: The IS domain, in both research and practice, suffers from an inadequate appreciation of risks associated with the complexity of advancing technologies [14]. This is evident by the scarcity of research output in understanding, measuring and mitigating a wide variety of risks in IS beyond the micro-management of specific risks. This can also be extrapolated on the basis of the high failure rate of software development projects [6].

The present research provides a novel contribution by creating a macro level framework of risks associated with information systems development and usage (ISDU). This paper examines a variety of perspectives on risk including those from the domains of IS, finance and operations management. The Standish Group report "CHAOS Summary 2009" showed that there was an increase in project failure rates and a noticeable decrease in project success rates. 32% of all projects succeeded with timely delivery, within budget completion and with required functionalities and features; 44% of the projects faced difficulties and these were late, over budget and (or) with compromised /incomplete functionalities and features; while 24% failed completely and were cancelled or delivered but could never be deployed /used.

For illustration purposes concerning risk management in financial services, here is an interesting quote: *"It's difficult to do risk assessment in this environment because of the added level of complexity involved"*. Marios Damianides [40] made this statement while serving as international president of the Information Systems and Audit Control Association (ISACA) and the IT Governance Institute following serious attempts to analyze the sub prime crisis. He reiterated that risk management technology has not been able to keep up with top global financial firms who have been introducing increasingly

J. Samuel is with Baruch College, City University of New York, New York, NY 10010 USA (email: jim.samuel@baruch.cuny.edu).

[1] A. Maynard. "New models needed to master technology trends – World Economic Forum," 2011. [Online]. http://2020science.org/2011/10/10/new-models-needed-to-master-technology-trends-world-economic-forum/

esoteric investment devices and variations of financial instruments such as CDOs (Collateralized Debt Obligations) and sophisticated equity derivatives. The value of these instruments is derived, at least in part from equity securities, using complex mathematics and software sophisticated programming, supported by high speed trading enabled computer systems. This example of the development of technology driven complexity is extensible: just as with the domain of finance, every other domain has leveraged the power of information systems by developing and deploying appropriate software systems. The disquieting fact is also that many of these systems, both in the private sector as well as in the public sector, are subject to an unexpectedly high probability of failure at various stages of their lifecycles - An IBM global CEO study (2008, a study involving 1500 global practitioners, conducted with ZEM, Center For Evaluation and Methods – University of Bonn, Germany) indicated that only a meager 41% of its projects were fully successful, implying a whopping 59% failure rate for projects across the world for projects related to any significant level of change.

These and other market events serve as a compelling call for IS researchers and practitioners to examine the domain of ISDU and articulate integrative risk models which, in the very least, must provide a conceptual identification of all major risk dimensions and risk types.

II. SCANNING THE RISK ECOSYSTEM

Integrated risk models must provide clear perspectives of various levels of risk, including, but not limited to: domain risk, process risk, methodology risk, technology risk, resource and security threats. In turning to IS literature to examine this phenomena, it is observed that in the past 25 years of the top two journals in IS. MIS Quarterly publications (1986 to Spring of 2012) had only 11 papers have used the word 'risk' in the title and about 8 more have "risk" in subject terms, taking the total to 18. Similarly there are about 10 articles with "risk" in the title and subject terms combined in Information Systems Research (1995 to 2012). Of these 29 articles in about 40 publication years between the two top IS journals, the primary emphasis has been on intrinsic systems risks and related process risks (Table 1:1). All the papers have inward looking implications for individual entities and none of these explore integrative or interactive models of risk with respect to the emerging and rapidly changing global technological environment. Interestingly, searching for "security (threat and failure management)" brought up more papers (Table 1:1) than that for "risk" in each of these top IS journals - again highlighting the micro-perspective bias in the choice of topics for research in IS.

There is also work on risk management, which simply appears to apply operations management principles in an adaptive and iterative manner to provide a measure of protection against obvious clusters of commonly respected threats. It appears that IS researchers have been making some scattered progress in developing disconnected pieces of knowledge associated with the potential weaknesses, flaws and intrinsic vulnerabilities - risk posed by ISDU and have unwittingly failed to study the big picture using a "bird's-eye view" strategy. Therefore, the present research direction aims to fill that gap by creating an expanded and integrative macro-framework for ISDU, with the intention that this expanded and integrative risk framework would possess better risk identification capability and explanatory power with regards to the wide array of risks facing ISDU.

Thus far the general perspective has been that IS risk is associated with the probability of something going wrong in some way through vulnerability, flaw or failure [64]. In the present study, I develop logical arguments using a multi-domain literature review strategy and an analysis of multiple perspectives on risk identification, risk measurement, risk mitigation, risk valuation, risk creation, risk management and risk control. These have been instrumental in shaping the call for integrative perspectives on risk presented in this paper. This paper leverages a inductive logic approach with phenomenological underpinnings as a research strategy to develop an integrative perspective model for risks associated with ISDU

A. *The Changing Nature of Risk*

Researchers and practitioners have hitherto placed a stronger emphasis on the rewards and provided attention to risk primarily to the unavoidable extent it would be required to preserve the survival of visible-return-on-investment-economics, explain obvious challenges to specific technology artifacts, tool imperfections and volatile processes. The advent of complex adaptive business systems (CABS) have resulted in evolving complexity [67] as we have leveraged IS to address a variety of problems, create value and improve the quality of life. We have reaped the rewards of the collective effects of the world adopting information technology and this societal force was defined by our needs, we created the rules. However, we have not acknowledged the iterative [25] deep structures [21] between these rules we have created and nor have we gauged the impact of the new and hitherto theoretically undefined risks. Giddens work on social theory [26] and his elaboration on structuration theory presents an interesting principle for exploring the relationship between IS and risk.

Structuration theory presents an evolutionary iterative path of "production and reproduction of actors and systems across time and space" using a helix structure and this perspective of reality can be applied to initiate the study of the risk in Information Systems by treating it as a societal effect, as elaborated below. The extraordinary capabilities of IS have kept our attention largely focused on the positive results of leveraging information technologies. Our action upon the IS "societal force" is our demand for rewards and benefits and as we receive them, we in our utilization of the benefits enter into the IS environment which is evolutionary in nature and presents risk cycles of increasing cross domain complexity and magnitudes.

The application of structuration theory to the study of information systems was popularized by Orlikowski's structurational model of technology [54] and also it

application to information technology and organizational life [53]. An extension of structuration theory is evident in the 'Adaptive Structuration Theory' (AST), which posits that technology and social processes act upon organizational change through the adoption of advanced technologies [17].

B. Measuring Macro-level Risk?

In a unique presentation of strategic thoughts, Michael Vitale writes about "The growing risks of information systems success" [70]. The paper describes "the risks of Information Systems success achieved in the absence of appropriate regard for the potential impacts." The paper provides a case study on how IS adoption and leverage produced competitive advantage. However, an inability to foresee the adverse effects of this success led to failure implying the absence of recognition of associated IS risks. The paper provides a theoretical discussion on how companies need to go beyond the obvious and explore the effects of IS adoption and leverage on a strategic and longer time horizon level. Vitale's model is parsimonious but highly conceptual and does not suggest a conscious recognition of risk beyond the call for the consideration of a longer time horizon. One of the present expressions defining IS security based on risk management models [59] are given by the conceptual equation "Security risk = (likelihood of security breach) x (cost of security breach)" and also "Security risk = (security breach rate) x (average cost per breach)". These models, though parsimonious, oversimplify the notion of IS risk to an extent which is counterproductive because none of the variables used have established measurement parameters nor is there any integrated and standardized comparative framework.

In recent IS research, there has been an attempt to adopt and integrate risk and risk relevant concepts from multiple disciplines into IS risk perception and management frameworks such as the work on IS security [43] which brings in concepts from system dynamics, cybernetic theory and Technological Threat Avoidance Theory (TTAT). Others have attempted to adopt singular concepts of risk from other disciplines in an attempt to develop improved perspectives and models for understanding and managing risks in IS projects such the work by Koch S [36] who argued for using "Value-at-Risk" which is a risk measure from the discipline of Finance for "IS/IT Project and Portfolio Appraisal and Risk Management." Other recent singular risk measure that have been adopted include infrastructure risk management concepts [49] and another economic concept of return on investment [4] has also been used a lens to view IS risks. However, there has been very little done to date to develop a conceptual framework that can serve an umbrella to house these various efforts within a systematic macro-perspective. A notable attempt in this direction is seen in the generic 'Project Management Body Of Knowledge' [55], commonly called as the PMBOK, which outlines key principles of risk management. However, the PMBOK fails to include domain risks with la levels of analysis perspective and thus in spite of a few macro-level factors, the primary focus is to start from multiple risk-identification points and travel downstream to focus on micro-level risk management techniques.

Classification	Journal:		
	ISR, 1995-2012	MISQ, 1986 - 2012	Combined
ROI Risk / IS Economics	2	1	3
Systems Risks	3	8	11
Information Risks	3	1	4
Process Risks	2	8	10
Risk of IS Success / Strategy		1	1
Total for "RISK"	10	19	29
Security / Threat/ Miscellaneous	13	30	43

Table 1.1: "Risk" in Title or Subject for Main IS journals

C. Literature on risks in IS & IS Risk Management

The four classical approaches to risk management in IS include Alter & Ginzberg's [2] implementation approach, Boehm's [8] software risk approach, McFarlan's [41] portfolio approach and Davis' [16] eventuality approach. Alter and Ginzberg [2] studied ways of managing uncertainty in IT execution by leveraging the change management model used by Klob and Frohman [35]. Interestingly they identified eight key risk drivers and their approach is one of deploying tactics to mitigate the risk factors and thus manage overall risk. Arguments positing that non-IT persons would find it difficult to understand IT deliverables due to the non-tangible nature of software were presented by Boehm [8]. He also argues that organizations tend to acquire newer technologies without evaluating all the associated risks as they cultivate impractical notions about the adaptability and flexibility of software. Boehm argues in favor of early detection and states that active management of risk will reduce failure and improve effectiveness. He also developed a two dimensional Risk Management typology risk assessment and risk control which had further sub-divisions.

Another prominent contribution is made by Barki, Rivard and Talbot [5] who posit that information communications technology (ICT) projects need to be appropriately controlled for the reduction of risks associated with ICT projects. They examined various issues and highlighted the absence of a systematic approach to rein in costs, meet user requirements and maintain project schedules by studying project with some level of failure including the "Allstate Insurance Company" case in which the cost estimates for a new information system changed from $8 million to about $100 million and from an original estimate of five years duration to an actual time of nine years.

Mark Keil has written extensively on risk in IS and one of his papers explores the dimensionality of risk [71]. This provides very interesting insights into the nature of risk management from an operational perspective but this work explores the dimensionality of risk within the context of IS project management and misses an opportunity to move the

analysis into the nature of risk itself and thus lacks ontological value. However, the arguments presented therein using sociotechnical systems theory demonstrate that "*social subsystem risk influences technical subsystem risk, which, in turn, influences the level of project management risk, and ultimately, project performance.*" Another work of significant importance was by McManus [42] who posited that "*the major causes of project risk as lack of planning and lack of top management control during the project life cycle.*" McManus proposed that IS projects tend to be started with some relevance to change and therefore such initiatives tend to be different from the ordinary and incremental change processes.

His Risk Management cycle approach consisted of four key phases, such that each phase must be performed and, repeated as necessary so as to optimally reduce risk and these 4 phases include "Establish that a risk exists; Analysis of risk severity and associated probability; Plan to manage the risk using the risk's severity and probability; Minimize risk consequence." He appears to have based this from Edward Deming's quality cycle with four phases Plan, Do, Study, Act. McManus also emphasizes the role of interactive forces in amplifying risk in software development.

Risk management can be described as a set of steps used for identifying, analyzing, measuring and controlling risk through the life of any project under consideration to meet its objectives [58]. Redzic et al. [56] tried to investigate, analyze and ascertain planned changes that considerably increased the software quality of all software products over a period of two years using the Six Sigma DMAIC approach, which is used for software quality improvement.

Current research [22] shows that most of the risk management initiatives in practice and also most of the academic research on risk management has focused on addressing commonly recognized challenges in efficiency, information security, project management and governance. They argue for the significance of the risks identified in the entire lifecycle of the ISDU. A risk management approach for building confidence and trust for Internet users is studied by Flinn and Stoyles [23]. Iversen, J. H., Mathiassen, L., and Nielsen, P. A. [32] use an action research strategy to develop an insightful model for risk management and their discussions on risk addresses various methodologies and tactics to mitigate risk. They however do not attempt to expand our view of risk itself and provides not insights into the scope of risks that could hold potential uncertainties and values at risk for SA & D projects.

Most of the risk management literature has thus far focused on individual aspects or a set of characteristics of ISDU risk but have, to the best of my present knowledge, failed to provide an integrative perspective, which combines various risk concepts into a single framework such that it is better represents the environment of risks for ISDU projects. . From a practitioner perspective, risk is reduced to elements that introduce uncertainty, quality and project schedule issues. However the present paper identities and integrates risk concepts relevant to ISDU on a higher level of overall relevancy than the past efforts.

III. THE DECOMPOSITION AND NATURE OF RISK

Detmar Straub and Richard Welke define risk as "Risk is the uncertainty inherent in doing business; technically, it is the probability associated with losses (or failure) of a system multiplied by the dollar loss if the risk is realized" [64]. Furthermore they extend the definition of risk specific to information systems ""Systems security risk is the risk that the firm's information and/or information systems are not sufficiently protected against certain kinds of damage or loss—is one form of systems risk. Another is project risk, the risk that a systems development project will fail." A clear consideration of risk has been viewed as necessary for understanding the impact of IT on economic organization [11]. This idea of risk is further elaborated and decomposed to include 'opportunism risk' and 'operations risk' [12]. They suggest the decomposition: "transactions cost = coordination cost + operations risk + opportunism risk." This decomposition of risk is insightful and useful. However, the scope of risk addressed is severely limited and therefore more holistic models are required to position this decomposition in the right perspective.

Another common understanding is that IS reduces risk through aiding transparency and price discovery [7]. However, additional research has shown that while this may be true on a quantitative count level of increased number of users who have access to market data, it may not be qualitatively true of all industries. Examining the online pricing in the computer industry, research has suggested that the qualitative aspect of price discovery may be in question: "IT-enabled online markets have clearly increased market transparency in terms of the accessibility and availability of price information. However, increased market transparency may not be directly translated into consumer benefits" [48].

A. Decomposing Risk

The financial domain has given much attention to risk and developed strong empirical models for risk identification and risk mitigation. One basic approach involves the decomposition of risk associated with any given equity into the sum of market risk and equity specific risk. Market risk is composed of stock prices, interest rates, foreign exchange rates, and commodity prices. This means that generic market level variance in any of these can prove to be a risk factor to the value of the equity under consideration without any fundamental change in the business value of the equity itself. Equity specific risk captures the risk inherent in that specific equity's business fundamentals. The third kind of risk that the financial domain refers to is 'systemic risk', not to be confused with 'systematic risk,' which is the same as 'market risk' discussed above. Systemic risk refers to the probability of loss from a catastrophic event that could collapse the entire financial system. Market risk cannot be diversified and market participants, being aware of this price their return expectation accordingly. Equity specific risk can be managed through diversification. In information systems risk analysis, the general tendency has been to focus on systems risk, which is akin to focusing on the risk associated with a specific equity.

IS practitioners and responsible managers are also, in many cases, prepared to mange the equivalent of a systemic crisis such as a flood or some such dramatic and often-instantaneous disaster by using well-defined disaster management and recovery processes.

B. Behavioral Risk

A great measure of uncertainty can be attributed to potential human behavior and various theoretical studies have identified a broad range of behavioral risks and human agent risks. Agency theory [20] addresses the issue of conflicting interests in case of a principal who hires an agent to achieve the principal's objectives but a complete dedication to the principal's objectives may hinder the accomplishment of the agent's self-objectives. Here human motivation is critical and agency theory posits that if the agent's motives are not aligned with that of the principal then the principal will be at risk to the degree of non-alignment.

ISDU provides opportunities for agency problems to arise when projects are undertaken and executed in a distributed environment where the 'agents' who are implementing may have alternative objectives, such as the recording more man hours for financial gains, as compared to the client who will be working to a strict time-line. On the other hand it could be that the 'agent' is working to the time for the sake of completion and avoidance of contractual penalties but compromises on code quality in the process, once again putting the 'principal' at substantial risk. Numerous such scenarios could lead to such agency issues in ISDU, spawning significant risks, which cannot be ignored. Thus this behavioral driver of project risk on various dimensions poses a real potential return on investment problem, which needs attention from ISDU researchers and practitioners.

Additionally, moral hazard theorists [24][44] posit that individuals and entities will act in a manner that propagates risk when the situations are such that these individuals or entities do not have to bear the costs of the risks they create. The challenge in coping with moral hazards is that behavioral aspects may be difficult to observe – such behavior is often based on information asymmetry in this that a entity who responds to an offer made by another entity may have private information, which could be used by the responding entity to take undue advantage of the entity making the offer. This is logically obvious in software development projects, where the 'experts' with private information, which is often superior to the subjects expertise / knowledge, take advantage of the situation to procure a contract or agree to a particular pricing structure with prior awareness of creation of future benefits or the awareness of favorable transfer of risk in the future. The impact of information asymmetry can be further amplified in case of increased project technological complexity, distributed implementation environments, cross cultural teams and subjective contractual arrangements: in each of these cases and others, the human agents involved have an increased opportunity to behave in a manner that increases ISDU project risks on various dimensions by leveraging information asymmetry and exercising moral hazards.

In using "bounded rationality" to explain organization learning [62], Simon has explained well the underlying concepts of how human intent and human ability interact with interesting consequences. The limitations of human rationality, even in cases where human objective is aligned with the intended goals and objectives, leads to the creation and propagation of risks, indicating higher levels of uncertainty than that which could have been gauged in the absence of the consideration of bounded rationality dimensions.

This notion of bounded rationality, in an open-ended manner, connects to the concept of 'irrational exuberance' in economics and finance. Alan Greenspan, in his now famous 1996 address, said "...Clearly, sustained low inflation implies less uncertainty about the future, and lower risk premiums imply higher prices of stocks and other earning assets. We can see that in the inverse relationship exhibited by price/earnings ratios and the rate of inflation in the past. But how do we know when irrational exuberance has unduly escalated asset values, which then become subject to unexpected and prolonged contractions as they have in Japan over the past decade?"

Yale professor, Robert Shiller [60] picked this term 'irrational exuberance' and expounded upon it with a clear articulation of how risk can be misestimated through the irrational exuberance phenomenon, with significant consequences both to the individual investor as well to the markets at large.

In spite of the wide acceptance of the Efficient Markets Hypothesis, behavioral finance has made rapid strides and crafted a space for itself partially on the basis of an inability of the efficient markets approach to gauge the levels of risks plaguing markets across the world. Expanding on the understanding of the firm boundary [13], another work by Holmström, Bengt and John Roberts [30] use the 'hold up' problem to explain the economics and contractual engagements of firms. 'Hold-up' phenomena refers to situations where two parties could be mutually benefited by working together in a pareto optimal fashion but do not do so because one of them may fear loss of negotiating power and eventually a loss of a degree of profitability due to this loss of negotiating power. This would be particularly applicable to ISDU situations where whole or part of the work is outsourced – a vendor may not want to over-commit or move ahead with a relationship suspecting that mathematically optimal approaches may lead to loss of bargaining power and thus reduce profitability in the times ahead. Another aspect is 'shirking' [27], which refers to a willful avoidance of work by employees or contracted agents to an extent such as would create maximum ease for themselves at significant cost and risk to the employer or principal.

C. Decomposing Risk

A discussion on risk with a focus on ISDU would be incomplete without developing a clear perspective on ISDU strategy risk and underlying business strategy risk. It is important to consider strategic risks, Michael Porter described

risk thus: "Risk is a function of how poorly a strategy will perform if the 'wrong' scenario occurs" [45]. ISDU projects could be affected by the technology strategy being adversely affected or due to non-performance of the underlying business strategy that could call for significant changes. Many projects fail due to the choice of inappropriate technology – projects are initiated using programming languages which are assumed to be able to efficiently serve the project purposes but the technology could either become redundant or it could be discovered that the technology selected does not best suit the project needs [18]. Operational strategies such as outsourcing or the adoption of various strategic methodologies also needs to evaluated with regards to risk – any evaluation without due consideration of associated risk would tend to provide an incomplete picture of the situation.

Value at Risk (VAR) is another way to view and measure risk, which has been extensively used in finance and economics both as an area of research and as a valuable risk measure in practice. VAR is not a substitute for risk adjusted value frameworks nor is it a purely probabilistic or stochastic measure for risk, though it leans on these in its development to some extent. VAR is used to measure the potential loss in a scenario, though by design VAR measures the potential loss in value of a portfolio with risk, over specific time periods, with a stated confidence interval. As an example, if the VAR on a portfolio with risky assets is $1 million for a 30-day period with a 95% confidence level, this implies that there is just a 5% chance that the value of the portfolio with risky assets will drop more than $1 million over any period of 30 days.

Practitioners in a variety of ways have used VAR and the measure is used to imply a possible loss in value from "normal market risk" thus contrasting it with overall risk that is a sum of market risks and non-market risks. A VAR measure has implications at a strategic level and adapting the use of VAR for ISDU would provide both business and technology managers with an excellent and tested framework for evaluating potential losses and develop contingencies. VAR measure could be used to effectively develop alignment between the business side or the client side and the technology side or the developer side through a careful planning process thus providing increased stability to ISDU initiatives. Koch S [36] has used the VAR model for "IS/IT Project and Portfolio Appraisal and Risk Management" and the present study posits the extension of this risk measure to ISDU and to IS initiatives at large.

D. Risk, Success and Failure

Risk and uncertainty are not synonymous in a conceptual sense there could be elements of risk in uncertainty [34], [57]. The purpose of this note on risk, success and failure is to support the development of an understanding for research questions Q1 and Q2 – by discussing risk, success and failure with a focus on delineating certainty, risk and uncertainty we can develop a better understanding of risk concepts and ways in which an integrated risk framework could be developed. Risk is associated with the probability of an event while uncertainty with the information associated with an event: "Variability is a phenomenon in the physical world to be measured, analyzed and where appropriate explained. By contrast uncertainty is an aspect of knowledge." (Sir David Cox). 'Certainty' refers to an event with only one possible outcome [57] irrespective of the nature of the outcome, whether it be success or failure or a specific point in the failure to success range. However, risk refers to events with two or more possible outcomes [57] and thus represents the probability of failure (or another measure) and the chance that vulnerability (or another measure) will be exploited. If a matter is known, no matter how serious, no matter how negative, devastating or disastrous – if it can be ascertained with hundred percent certainty, then there is no risk [57], [9], [37], [66], there is only a certain failure or a guaranteed catastrophe that exists.

It must be noted that "no risk" is very different from "zero risk" or even "riskless". "No risk" in the present context simply implies that the outcome in specific and certain. Risk is associated with probability and therefore with non-guarantee-able, but known variability. Where uncertainty exists –there risk exists at least in perception but it is not necessary that uncertainty must exist with risk.

For example, if an individual jumps of an aircraft flying at 30,000 feet above sea level without any parachute or any alternative safety mechanism to the earth below, then there is no risk, only the certainty of death. However, if the same individual were to jump of an aircraft flying at 30,000 feet above sea level with a parachute hat has not been tested or used for a long time and is suspected to have been damaged in transportation or is know to have a design flaw which sometimes causes it to not open and works well otherwise, then the individual is taking a risk which is a function of at least two variables: one is the probability that the parachute won't open or work appropriately enough to protect the diver from a lethal fall and the second is the extent of damage or the value at stake, which in this case is the person's life. This is in line with past research where risk has been identified as being a function of the probability of an event and the extent of impact caused by or associated with the probable event [34]. The first case is sure death, and therefore the presence of certainty but no risk and the second case is probable death (more than one possible outcome) and hence the presence of risk, along with the implicit absence of certainty. This is in line with past research [57], [9], [37], [66], which classify the occurrence of certain events as being different from risky events. Additionally, this notion of 'no risk' is supported by research on decision making under risk which includes the Expected Utility [48] and Prospect theory [33] – the important takeaway being that "no risk" is assumed by a rational agent who would either optimize for certain success or avoid totally for certain failure. This helps us to understand that success involves the mitigation and avoidance of risk and using the same logical thought process as above, we can posit that it is possible and though weakly in certain domains, to have success without risk. It is also necessary to note the difference between risk perceptions [46][51] which are subjective or descriptive measures of risk and risk evaluation which include

objective measures of risk and are based on quantitative methods. From an implementation perspective, managers need to take decisions which are often associated with risky events and for the purposes of addressing research question 2 of the present paper, Utility theory [48] and Prospect theory [33] would serve to support the risk concepts integration framework.

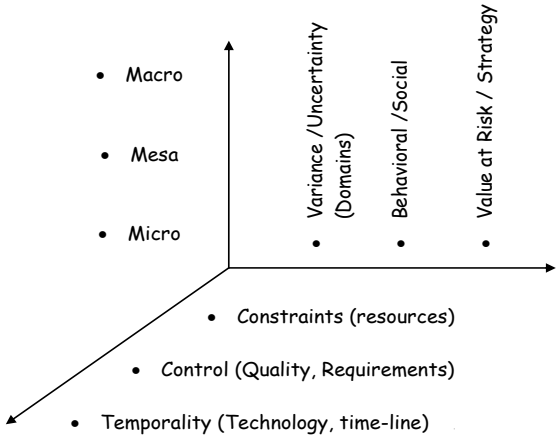
IV. Towards an Integrative View of ISDU Risks

It is important to reiterate the importance of the IT artifact in ISDU, and strong arguments for the same on a broader perspective on the IS research domain has been made by Orlikowski [52]. Here we acknowledge not only technological artifacts such as programming languages, necessary hardware, coding, methodologies and innovations but we also need to acknowledge the non-IT artifacts such as human resources, financial resources, organizational resources (those in addition to human and financial) and intangible assets, all of which interact with the IT artifact and ISDU processes to impact risk measures.

It is through this dynamic interaction of IT artifacts with non-IT artifacts that we see an increase in complexity levels, the usual effect of which should be an increase in uncertainty and thus normatively, but not necessarily, an increase in the overall ISDU process or project risk measure. This interaction-istic view of information systems is not novel and has been well addressed by Silver Mark S., M. Lynne Markus and Cynthia Mathis Beath [61] in their argument for the scope, content and pedagogical context for IS courses. They refer to the model they develop as the "Information Technology Interaction Model" and they posit it to be so because of their argument that information systems covers the interaction of technology with the organization, implicit here is the idea that technological artifacts are in interaction with organizational, a.k.a. business, artifacts and this interaction works in a similar fashion within the narrower scope of ISDU. This understanding of the interaction-ist nature of technological artifacts is necessary for the development of an expanded framework of risk concepts because the expanded framework is directly relevant to a broader involvement of technological artifacts.

Numerous risks plague ISDU projects and risks are commonly identified with the operational measures such as the risk of the ISDU project not being completed on time or the risks associated with the ISDU initiatives going over the allocated budget or the risks for the development objectives not being met, in part of the failure of the project as a whole. The present paper both expands the scope of risks taken into account for ISDU projects and also looks at risks as being multidimensional in the way they influence ISDU. The present investigation leads to the creation of new directional framework: "Conceptual Integrative Risk (CIR)" framework for ISDU initiatives – this framework is three dimensional in nature and incorporates three levels of analysis (Micro, Mesa & Macro) along the vertical axis. The two horizontal axes are used to represent risk with the forward axis representing risk measures (constraints of temporality, control and quality, and resources -including financial resources). The second and lateral horizontal axis captures the risk dimensions and includes risk constructs of variance and uncertainty which tend to be driven by technological and process factors, behavioral risks which play out in the social context and VAR which provides an overall view of the downside – the potential loss that the ISDU initiative owners would need to be prepared for.

In the past, many frameworks and risk management methodologies have been posited: Vepsalainen [69] and [38] refer to the four classical approaches to risk investigation and management, the essence of which has been carried over to modern day research. These historical and classical risk management approaches help us to understand the fragmented way in which risk in IS has been subsequently articulated. Alter, S. and Ginzberg, M. [2] address the issue of uncertainty in IS implementation scenarios but their notion of risk is limited to the probability of failure in the implementation process and cannot thus be extended to accommodate other issues such as budget risks and behavioral risks. Likewise, Boehm [8] restricts the analysis to a loss minimization approach in favor of the stakeholders and Davis et al. [16] focus on the risk of non-achievement of alignment based on his focus on the difficulties in understanding the task well enough through the requirements gathering process – again a good but very limited way of looking at risk. McFarlan [41] takes a more business centric perspective by focusing on the project's goals and develops the idea of risk around the probability of failure to meet all or some of the goals. In all of this and in practice, based on what we have seen so far and to the best of my present knowledge, no significant and scholarly work exists on risks in ISDU from an integrative and broad perspective.



Conceptual Integrative Risk (CIR) Framework for ISDU

Figure 1

It is against this backdrop of fragmented theoretical studies and a dismal success rate for IS projects that the present framework for gauging risk in a broad and encompassing manner is presented. This directional framework is presented

as a conceptual model "CIR" (Conceptual Integrative Framework).

Notes on the RCM qualitative methodology for the proposed Conceptual Integrative Risk (CIR) Framework: The development of the idea of an expanded and integrative risk framework "CIR" has been largely conceptual thus far and has limited applicability in its present form. Though it would be of theoretical interest to continue conceptual development and there would be significant value in articulating the ontological argument on the nature of risk in IS, it becomes necessary from a relevance perspective to develop this study using a qualitative methodology which supports conceptual development. The 'Revealed Causal Mapping' (RCM) method provides strong support for theoretical and conceptual development and is evocative in nature. Prior studies indicate that the use of the RCM methodology has been successful in various research streams including IS [47]. Also importantly, RCM serves as an evocative qualitative methodology, which reveals the relationships between the various theoretical constructs being explored. The present study represents an opportunity for the development of a mid-range theory in the development and usage of an expanded and integrative risk concepts framework and the RC methodology lends itself to the development of such mid-range theories [46].

V. CONTRIBUTION AND IMPLICATIONS

The present research on the development of an expanded and integrative cross-domain risk framework .is expected to thus stimulate thought and encourage integrative perspectives on risk in ISDU amongst researchers and practitioners:

1) Serve as an initiation of an ontological discussion on the nature, the span, the scope and the process of risk in information systems.
2) Provide an expansion of risk concepts associated with IS risk perspective frameworks starting with CIR
3) Serve as a integration point for cross-domain risk measures
4) Serve as a starting point for developing customized risk management models and solutions
5) Serve as a new stream of research on the application of cross-domain risk measures in IS
   a) Case studies to explore risks in IS projects
   b) Qualitative studies to analyze the integration of various risk measures
   c) Experimental studies to analyze the effects of expanded cross-domain risk frameworks upon project performance

From a practitioner perspective, the present research can be expected to add a new dimension to risk investigation, risk identification and risk management. The CIR framework for ISDU also provides a better explanation of the way risks emerge in a multi-contextual situation where ISDU activities could be simultaneously implemented on various parts of a system in an organization. The risks that arise of the resultant complexities can be understood using the Value at Risk framework while specific contextual risks may be best understood using variance and uncertainty constructs or by leverage the behavioral construct. Risk is a very relevant and high priority subject for individuals and corporations. The extraordinary impact of IS on day-to-day life and the high amount of latent risks involved need to be studied and carefully modeled so that we can minimize the impact of risks and maximize the rewards of technological advances in ISDU and IS initiatives at large. Specifically, in addition to the first five points, practitioners can also benefit from the CIR framework and the theoretical arguments supporting the same by:

6) Using the CIR framework for strategic risk analysis – the framework being conceptual in nature along with its ability to provide a macro-perspective on risks, can serve as a direction setting and strategic analysis tool.
7) Leveraging the CIR framework to support existing risk management models and allow for positioning of such existing or future risk management models into the broader CIR ecosystem.

VI. LIMITATIONS OF THE PRESENT STUDY

This research proposal is qualitative in methodology and significantly conceptual in its articulation. The nature and the scope of the research topic has mandated that a conceptual framework needs to be established to create a direction for empirical research. The exhaustive level of literature review that is mandated by a topic of this nature has not occurred – there remain large sections of literature related to risk concepts that need to be reviewed and incorporated into the further development of the CAR framework.

The present research proposal is expected to be high on external validity but the generalizability and the applicability of the model will remain a challenge until we see a stream of case studies, empirical and quantitative studies on the topic. Future studies need to explore empirical relationships between various constructs of IS capabilities and associated risks. Future research must also expand the scope and examine additional constructs such as systems rigidity and model variations for specific industries. The present research only touches upon one aspect of risk management and that is the identification of risk. There is significant scope for directing this research into a sub-stream of risk management.

VII. CONCLUSION

The present research proposal is expected to be a significant conceptual contribution to IS theory in expanding the understanding of risks associated with ISDU in a broad and encompassing framework which identifies the risk measures, dimensions and levels of analysis in an encompassing but parsimonious manner. This paper intends to emphasize that the commonly understood and used measures for identifying risks associated with ISDU projects are insufficient as they would tend to supply an incomplete and restrictive perspective on associated risks. Mere accounting of risk measures without

a simultaneous consideration and inclusion of risk dimensions would lead to an understatement of risks associated with ISDU. This understatement would not only provide a misleading application of the risk adjusted economic values but would also miss out on identifying certain risks altogether.

The present research is expected to create a shift from the general perspective that risk is measured and managing only by addressing commonly accepted risk measures, which are only the surface expressions of the underlying risk dimensions. Developing mathematical equations using econometric modeling to capture this multifaceted view of risks in ISDU would be useful though it is beyond the scope of the present paper.

We need to bear Anthony Giddens structuration principle in mind to recognize that the ISDU risk – rewards environment is dynamic, iteratively evolving and changing t any given point of time. This adds increased complexity to an already sophisticated mix. Today's "thought leadership" could be tomorrow's obsoletes. Therefore, the present contribution is expected to provoke and initiate a scholarly debate on risks associated with ISDU, with a broad encompassing perspective on risks, which not only includes commonly known and accepted risk measures but also includes risk dimensions that are critical for developing a fair, complete and holistic view of risks. This will be of significant help to both researchers and practitioners – researchers can use this to explain various risk phenomena better and articulate multi-dimensional risk management models while practitioners would be able to use this a starting point to better evaluate the scope and depth of risks, empowering them to create better risk management models and an advancement in understanding the economics of and ISDU projects and IS initiatives from a risk adjusted return perspective.

REFERENCES

[1] Alter, S. (2003) "18 Reasons Why IT-Reliant Work Systems Should Replace 'The IT Artifact' as the Core Subject Matter of the IS Field," Communications of the AIS, 12(23), October, pp. 365-394

[2] Alter, S. and Ginzberg, M. (1978). Managing Uncertainty in MIS Implementation, Sloan Management review, 20, 23-31.

[3] Alter, S., and Browne, G. J. (2005) "A Broad View of Systems Analysis and Design: Implications for Research" Communications of the AIS 16(5) 981-999.

[4] Armour, P. G., 2010, "The Business of Software – Return at Risk" Communications of the AGM, Sept 2010

[5] Barki, H., Rivard, S. and Talbot, J. (1993). Toward an assessment of software development risk, Journal of Management Information Systems, 10, 203.

[6] Bloch Michael, Sven Blumberg, and Jürgen Laartz; Delivering large-scale IT projects on time, on budget, and on value, McKinsey Quarterly -October 2012

[7] Bloomfield,R.,and O'Hara, M. "Market Transparency: Who Wins and Who Loses?," The Review of Financial Studies (12:1), 1999,pp. 5-35.

[8] Boehm, B. W. (1991). Software Risk Management: Principles and Practices, IEEE Software, 30, 32-40.

[9] Brännmark, Johan and Sahlin, Nils-Eric "Ethical theory and the philosophy of risk: first thoughts" Journal of Risk Research, Vol. 13, No. 2, March 2010, 149–161

[10] Checkland, P. (1997) Information, Systems, and Information Systems, Chichester, UK: John Wiley, cited in Rose and Meldrum (1999)

[11] Clemons, E.K., and Row, M.C. Information technology and industrial cooperation: the role of changing transaction costs. Journal of Management Information Systems, 9, 2 (Fall 1992), 9-28.

[12] Clemons, Eric K., Sashidhar P. Reddiand Michael C. Row, "The Impact of Information Technology on the Organization of Economic Activity: The "Move to the Middle" Hypothesis," Journal of Management Information Systems, 10, 2, (1993), 9 – 35 .

[13] Coase, R. "The Nature of the Firm," *Economica* , Blackwell Publishing, vol. 4, 16, 1937, pp. 386–405.

[14] Collins, Jon; New technology brings new risks, Computing 13612972, 4/24/2008

[15] Damianides, Marios, former international president of the Information Systems and Audit Control Association (ISACA) and the IT Governance Institute, and a partner in the Risk Advisory Services for Ernst & Young in New York : http://www.ciozone.com/index.php/Case-Studies/Subprime-Mess-What-Role-Did-IT-Playu.html

[16] Davis, G.B., A.S. Lee, K.R. Nickles, S. Chatterjee, R. Hartung, and Y. Wu, Diagnosis of an information system failure: A framework and interpretive process. ;In Proceedings of Information & Management. 1992, 293-318.

[17] Desanctis, G. & Poole, M. S. (1994). Capturing the Complexity in Advanced Technology Use: Adaptive Structuration Theory. Organization Science. 5, 121-147

[18] Dorsey, P; "Top 10 Reasons Why Systems Projects Fail" Harvard – HKS publications, 2005.

[19] Dynamic Contracts with Moral Hazard and Adverse Selection.Full Text Available By: Gershkov, Alex; Perry, Motty. Review of Economic Studies, Jan2012, Vol. 79 Issue 1, p268-306,

[20] Eisenhardt, K. (1989) Agency theory: An assessment and review, Academy of Management Review, 14 (1): 57-74.

[21] El Sawy, O. A. 2003. The three faces of information systems identity: Connection, immersion, and fusion. Comm. Assoc. Inform. Systems 12 588–598.

[22] Ezamly Abdelrafe and Burairah Hussin, International Management Review, Vol. 7- No. 2, 2011

[23] Flinn, S., & Stoyles, S. (2004). Human factors: Omnivore: Risk management through bidirectional transparency. Proceedings of the 2004 workshop on new security paradigms, 97-104

[24] Alex Gershkov & Motty Perry, "Dynamic Contracts with Moral Hazard and Adverse Selection," *Review of Economic Studies,* Oxford University Press, vol. 79, 1, 2012, pp. 268-306.

[25] Giddens, A. (1986). Constitution of society: Outline of the theory of structuration, University of California Press; Reprint edition (January 1, 1986)

[26] Giddens, Anthony (1979) Central problems in Social Theory : Action, Structure and Contradiction in Social Analysis. London : Macmillan.

[27] Gintis H.(1976):"The nature of labor exchange and the theory of capitalist production", Review of Radical Political Economics, 8 (2), p36–54.

[28] Glaser, B. (1992). Basics of grounded theory analysis. Mill Valley, CA: Sociology Press.

[29] Greenspan, Allan. "The Challenge of Central Banking in a Democratic Society", 1996-12-05

[30] Holmström, Bengt,& John Roberts (1998). "The Boundaries of the Firm Revisited," Journal of Economic Perspectives, 12(4), pp. 73-94

[31] Iivari, J., J. Parsons, and A. R. Hevner (2005) "Research in Information Systems Analysis and Design: Introduction to the Special Theme Papers," Communications of the AIS, 16, pp. 810-813.

[32] Iversen, J. H., Mathiassen, L., and Nielsen, P. A. (2004) "Managing Risk in Software Process Improvement: An Action Research Approach" MIS Quarterly 28(3): 395-433.

[33] Kahneman, Daniel and Tversky, Amos; "PROSPECT THEORY: AN ANALYSIS OF DECISION UNDER RISK" Econometrica (pre-1986); Mar 1979; 47, 2;

[34] Kaplan, Stanley and B. John Garrick "On The Quantitative Definition of Risk" Risk Analysis, Vol. I, No. I, 1981

[35] Klob, D. A. and Forhman, A. L. (1970). An Organizational Development Approach to Consulting, Sloan Management Review, 51-65.

[36] Koch S (2006) "Using Value-at-Risk for IS/IT Project and Portfolio Appraisal and Risk Management" The Electronic Journal Information Systems Evaluation Volume 9 Issue 1, pp 1-6, available online at www.ejise.com

[37] Lapin, Lawrence L. and Whisler, William D., "Quantitative decision making with spreadsheet applications"D uxbury/Thomson Learning, 2002 - Business & Economics Book
[38] Lucas, H. "Implementation—The Key to Successful Information Systems," New York, Columbia University Press, 1981.
[39] Malone, T.W.; Yates, J.; and Benjamin, L Electronic markets and electronic hierarchies. Communications of the ACM, 30, 6 (June 1987), 484-497.
[40] Marios Damianides, former international president of the Information Systems and Audit Control Association (ISACA) and the IT Governance Institute, and a partner in the Risk Advisory Services for Ernst & Young in New York : http://www.ciozone.com/index.php/Case-Studies/Subprime-Mess-What-Role-Did-IT-Playu.html
[41] McFarlan, F. Warren; "Portfolio Approach to Information Systems" HBS reprint 1981.
[42] McManus, J. (2001). Risk in Software Projects, Management Services, 45, 6-10.
[43] Mejias, Roberto J. "An Integrative Model of Information Security Awareness for Assessing Information Systems Security Risk," IEEE Conference proceedings, 2012, 45th HICSS.
[44] Mirrlees, J. A. 1999. "The Theory of Moral Hazard and Unobservable Behaviour: Part I," in *Review of Economic Studies,* Wiley Blackwell, vol. 66, issue 1, 1999, pp. 3-21.
[45] Michael E. Porter, *Competitive Advantage: Creating and Sustaining Superior Performance*, Free Press, 1998 (1985)
[46] Narayanan, V. K., and Fahey, L. "Evolution of Revealed Causal Maps During Decline: A Case Study of Admiral," in Mapping Strategic Thought, A. S. Huff (ed.), John Wiley & Sons, Chichester, UK, 1990, pp. 107-131.
[47] Nelson, K.N.; Nadkarni, S.; Narayanan, V.K.; Ghods, M. (2000),"Understanding software operations support expertise: A revealed causal mapping approach". MIS Quarterly 24(3): 245-277.
[48] Neumann, John von and Morgenstern, Oskar; Theory of Games and Economic Behavior. Princeton, NJ. Princeton University Press. sec.ed. 1947.
[49] Öbrand, L., Augustsson, N-P., Holmström, J. & Mathiassen, L, "The Emergence of Information Infrastructure Risk Management in IT Services," 2012. Accepted to *HICSS-45*.
[50] Oh Wonseok and Henry C. Lucas, Jr., MIS Quarterly Vol. 30 No. 3. pp. 755-775/September 2006
[51] Oltedal,Sigve and Bjørg-Elin Moen, Hroar Klempe, Torbjørn Rundmo; "Explaining risk perception. An evaluation of cultural theory" c Rotunde publikasjoner, Rotunde no. 85, 2004
[52] Orlikowski Wanda J & Iacono Suzanne C,"Desperately seeking the "IT" in IT research - A call to theorizing the IT artifact." Information Systems Research, Vol12, No.2, June 2001, P 122- 134;
[53] Orlikowski, W.J. and Robey, D. "Information Technology and the Structuring of Organizations," Information Systems Research, Vol. 2(2), June 1991
[54] Orlikowski, W.J."The duality of technology: Rethinking the concept of technology in organizations. 1992, Organ Sci. 3(3) 398-427.
[55] PMBOK - PMI (2012), A Guide to the Project Management Body of Knowledge, 5th Ed.
[56] Redzic, C., & Jongmoon, B. (2006). Six Sigma Approach in software quality improvement. Fourth International Conference on Software Engineering Research, Management and Applications, 396
[57] Riabacke, Ari Riabacke;;"Managerial Decision Making Under Risk and Uncertainty" December 2006, IAENG International Journal of Computer Science, 32:4, IJCS_32_4_12
[58] Schawlbe, K. (2005). Information technology project management Fourth Edition, Course Technology: Thomson.
[59] Schechter, Stuart E; Harvard University-http://www.eecs.harvard.edu/~stuart/papers/eis04.pdf
[60] Shiller, Robert; Irrational Exuberance - Princeton University Press 2000, Broadway Books 2001, 2nd ed., 2005
[61] Silver Mark S., M. Lynne Markus and Cynthia Mathis Beath, The Information Technology Interaction Model: A Foundation for the MBA Core Course; MIS Quarterly, Vol. 19, No. 3, Special Issue on IS Curricula and Pedagogy (Sep., 1995),pp. 361-390
[62] Simon, Herbert (1991). "Bounded Rationality and Organizational Learning". Organization Science 2 (1): 125–134
[63] Standish Group report "CHAOS Summary 2009"
[64] Straub, D. & Welke, R. "Coping with Systems Risk: Security Planning Models for Management Decision-Making, MIS Quarterly, (22: 4, December), 1998, pp. 441-469.
[65] Strauss, A. (1987). Qualitative analysis for social scientists. Cambridge, England: Cambridge University Press.
[66] Taha, Hamdy; Operations Research (1987, Book, Illustrated) Mcmillan publishers, Taha|ISBN-10: 0024189405 | ISBN-13: 9780024189400
[67] Tanriverdi, Huseyin, Arun Rai, and N. Venkatraman. 2010. "Reframing the Dominant Quests of Information Systems Strategy Research for Complex Adaptive Business Systems." Information Systems Research 21 (4) (November 18): 822-834.
[68] The Theory of Moral Hazard and Unobservable Behaviour: Part I.Full Text Available By: Mirrlees, J. A.. Review of Economic Studies, Jan99, Vol. 66 Issue 226, p3-21
[69] Vepsäläinen T. Saarinen, A.; Managing the risks of information systems implementation, Journal: European Journal of Information Systems - EJIS , vol. 2, no. 4, pp. 283-295, 1993
[70] Vitale, Michael R., MIS Quarterly; Dec86, Vol. 10 Issue 4, p327-334, 8p
[71] Wallace, L., Keil, M., and Rai, A., "How Software Project Risk Affects Project Outcomes: An Investigation of the Dimensions of Risk and an Exploratory Model," Decision Sciences, Vol. 35, No. 2 (Spring), 2004, pp. 289-321

AUTHOR BIOGRAPHIES

Ben Andrews is a graduate student in the Department of Computer Security at the Rochester Institute of Technology.

Kallol K. Bagchi holds the Nita and Jim Phillips endowed Professorship in Information and Decision Sciences in the College of Business at the University of Texas at El Paso. He received a Ph.D. in Business from Florida Atlantic University and a Ph.D. in Computer Science from Jadavpur University, India. He is the author of over 20 articles in such journals as Communications of the AIS, Communications of the ACM, Journal of Information Technology & Decision Making, and Information and Management. He curreently serves as Editor-in-Chief of the Journal of Information Privacy and Security.

Richard Ballard is a student in the Department of Computer Science at Rochester Institute of Technology.

Ernst Bekkering is an Associate Professor in Information Systems at Northeastern State University in Tahlequah, OK. He has published papers and presented at conferences in the areas of information systems security and information system user perceptions.

Zachary Birnbaum is a PhD student at Binghamton University. His research interests include behavior based intrusion detection, graph processing, data mining.

Gehana Booth is currently finishing up her B.C.S. at Carleton University in Ottawa, Canada. She will be continuing her studies at Carleton next year in starting her M.C.S. with the Carleton Computer Security Lab. Her research interests include implicit authentication, anomaly detection, and security on non-standard platforms such as mobile platforms and the cloud.

Stéphane E. Collignon is a Ph.D. candidate and an instructor in the Business Information Technology Department at Virginia Tech. His current research interests are in the areas of privacy in information technology, logistics, and agent based modeling. He received a BBA from Institut Commercial de Nancy – France, and a MBA from Duquesne University – Pittsburgh, PA.

Trushank Dand is a graduate student in the Department of Computer Science at the Rochester Institute of Technology.

Amit V. Deokar is an Associate Professor of Information Systems in the College of Business and Information Systems at Dakota State University. His primary research interests are in decision support systems/analytics, and business process management, along with application areas such as healthcare informatics. He has published in journals such as Journal of Management Information Systems, Communications of the AIS, and The DATA BASE for Advances in Information Systems, and has presented his research at a number of conferences including ICIS, AMCIS, HICSS, and DSI. He holds a BE in Mechanical Engineering from V.J. Technological Institute, Mumbai, a MS in Industrial Engineering from the University of Arizona, and a PhD in Management Information Systems from the University of Arizona. He is a member of AIS, MWAIS, ACM, and AAAI, and can be reached at amit.deokar@dsu.edu.

Andrey Dolgikh is a PhD student at Binghamton University. His research interests include behavior based intrusion detection, graph compression, colored petri nets.

Rishabh Dudheria is currently pursuing the Ph.D. degree in Electrical and Computer Engineering at Rutgers University. He received the M.S. degree in Electrical and Computer Engineering from Rutgers University in 2008

Omar El-Gayar, Ph.D. is a Professor of Information Systems and Dean of Graduate Studies and Research, Dakota State University. His research interests include: decision support systems, multiple criteria decision making, and the application of decision technologies in healthcare, environmental management, and security planning and management. His inter-disciplinary educational background and training is in information technology, computer science, economics, and operations research. Dr. El-Gayar's industry experience includes working as an analyst, modeler, and programmer. His numerous publications appear in various information technology-related fields. He is a member of AIS, ACM, INFORMS, and DSI.

Kimberly Francis is a student in the Department of Computer Science at Rochester Institute of Technology. She also currently works as a web and application developer for Datto Inc.

Sanjay Goel is an Associate Professor in the Information Technology Management Department (School of Business) at the University at Albany, SUNY. He is also the Director of Research at the New York State Center for Information Forensics and Assurance at the University. Before joining the university, he worked at the General Electric Global Research Center. His current research interests include self-organized systems for modeling of autonomous computer security systems using biological paradigms of immune systems, epidemiology and cellular regulatory pathways. He also actively works on distributed service-based computing, network security and active networks. His research includes use of machine learning algorithms to develop self-learning adaptive optimization strategies and use of information theoretic approaches for classification of data for use in applications such as portfolio analysis and information assurance.

Wencui Han is a candidate for the doctoral degree in Management Science and Systems at the State University of New York, Buffalo, NY. Her research interests include Communications and Compliance, Disaster Preparedness and Response Management, Information Assurance and Patient Safety and healthcare Delivery Systems.

Tabitha L. James is an Associate Professor in the Business Information Technology Department at Virginia Tech. Her current research interests are in the areas of security and privacy in information technology, combinatorial optimization, heuristics, parallel computing, and social networks. She has published in journals such as IEEE Intelligent Systems, IEEE Transactions on Evolutionary Computation, European Journal of Operational Research, Computers and Operations Research, Computers & Security, and Engineering Applications of Artificial Intelligence. She received a BBA in Management Information Systems and a Ph.D. in Business Administration from the University of Mississippi.

Carl A. Janzen is a Sessional Instructor in the Computer Science Department at the University of the Fraser Valley (UFV). He has a BCIS from UFV and a MSc in IT from Liverpool.

Byung Cho Kim is an Assistant Professor of Logistics, Service and Operations Management at Korea University Business School. He received his Ph.D. in Industrial Administration from the Tepper School of Business of Carnegie Mellon University. Before joining Korea University, he served as an Assistant Professor of Business Information Technology at the Pamplin College of Business of Virginia Tech. His research has appeared in prestigious academic journals including Production and Operations Management, Computational Economics, International Journal of Electronic Commerce, and Decision Support Systems.

Peeter J. Kirs is an Associate Professor of Information and Decision Sciences in the College of Business at the University of Texas at El Paso. He holds a Ph.D. and an M.B.A. from the State University of New York at Buffalo and has published over 20 articles in such journals as Management Science, Decision Sciences, Management Information Systems Quarterly (MISQ),

CACM and Information & Management.

Liyun Li is a Ph.D candidate in the Computer Science Department at Polytechnic Institute of New York University, where he is under the supervision of Prof. Nasir Memon. His research interests include areas of Data Mining, Machine Learning, Cyber Security, Network Measurement and Computer Forensics. He is currently collaborating with the AT&T Security Research Center on a research project to detect P2P communities on the Internet. He is also interested in developing network security tools and has been actively collaborating with the IBM Watson Research Security Group on developing a network Reputation System to discover infections. He is also interested in development of mathematical models, especially financial risk models.

Nasir Memon is a professor in the Department of Computer Science and Engineering and director of the Information Systems and Internet Security (ISIS) laboratory at NYU-Poly. He is one of the founding members of the Center for Interdisciplinary Studies in Security and Privacy (CRISSP), a collaborative initiative of multiple schools within NYU including NYU-Steinhardt, NYU-Wagner, NYU-Stern and NYU-Courant. His research interests include digital forensics, data compression, and multimedia computing and security. Memon earned a Bachelor of Engineering in Chemical Engineering and a Master of Science in Mathematics from Birla Institute of Technology and Science (BITS) in Pilani, India in 1981. He received a Master of Science in Computer Science and a PhD in Computer Science from the University of Nebraska. Prof. Memon has published over 250 articles in journals and conference proceedings and holds a dozen patents in image compression and security. He has won several awards including the National Science Foundation's CAREER Award and the Jacobs Excellence in Education Award from NYU-Poly. He has been on the editorial boards of several journals and was the Editor-In-Chief of Transactions on Information Security and Forensics. He is an IEEE Fellow and a distinguished lecturer of the IEEE Signal Processing Society. Memon is the co-founder of Digital Assembly and Vivic Networks, two early-stage start-ups in NYU-Poly's business incubators.

Phil Menard is a second-year doctoral student in business information systems at Mississippi State University. He is particularly interested in security education training and awareness (SETA) programs and the impact of espoused cultural values. He has published and presented at the IFIP Dewald Roode Workshop on Information Systems Security Research and has served as a reviewer for the European Conference of Information Systems and the Americas Conference of Information Systems. His primary research is focused on behavioral IS security issues.

Dr. Richard P. Mislan is currently serving in the Dept. of Computing Security at Rochester Institute of Technology (RIT), his alma mater from 1991. In his capacity there, he has been charged with developing the new Mobile Device Security and Exploitation program as part of the new Department of Computing Security in the Golisano College of Information Science and Computing. He has a Ph.D. in Information Systems, from Nova Southeastern University, a M.S. degree in Information Systems Management from Ferris State University, and a B.S. in Professional and Technical Communications from the Rochester Institute of Technology. He has over 20 years of experience in information systems, security, and forensics as an educator and researcher. Before he joined RIT, he was an Assistant Professor and CERIAS researcher at the Cyber Forensics Lab at Purdue University's College of Computer and Information Technology, where he was leading the advancement of small scale digital device forensics through such creations as the Mobile Forensics World conference and the Small Scale Digital Device Forensics Journal. Mislan previously served in the United States Army as an Electronics Warfare

Officer.

Arunabha Mukhopadhyay, Ph.D is an Associate Professor of Information Technology & Systems Area at Indian Institute of Management Lucknow (IIM Lucknow). His research interests include IT Risk Management, Quantifying IT Risk, Cyber-risk insurance, IT Governance, IT Audit, Network Security, Healthcare IT, Network Science, Data mining, e-governance and Telecom Management. He has co-supervised 3 doctoral theses and published around 40 papers in various referred journals and conferences including *DSS, JIPS, IJISCM, Decision, IIMB Review, CSI-C, HICSS, AMCIS, Pre-ICIS workshops, GITMA, CISTM, ICEG* etc. He is the recipient of the *Best Teacher in Information Technology Management* in 2013 and 2011, by Star - DNA group B-School Award and 19th Dewang Mehta Business School Award, in India respectively.
He is a Member of *IEEE, AIS, ISACA, DSI, ITS, IFIP WG 11.1* and a Life Member of Computer Society of India (CSI), Telemedicine Society of India (TSI), Indian Insurance Institute (III), Actuarial Society of India (ASI), All India Management Association (AIMA), System Dynamics Society of India (SDSI) and, Operations Research Society of India (ORSI) .
He has obtained his Ph.D. and Post Graduate Diploma in Business Management (PGDBM) from Indian Institute of Management Calcutta (IIM Calcutta), in the area of Management Information Systems. He was awarded the *Infosys scholarship* during his Ph.D.

Michael Oehler is an information assurance professional with twenty-five years of experience in computer and network security. He has contributed to the design of various IEFT security protocols, most notably IP Security (IPSEC) and secure DNS, lead research in the area of IP traceback, and written about VoIP security. As a network defender, Mr. Oehler has performed network anomaly detection at line-speed, attenuated adversarial activity in a defensive environment, and is currently defending our networks against the Advance Persistent Threat (APT.) He is pursuing his PhD at the University of Maryland, Baltimore County, and the realization of a private packet filtering language. Mr. Oehler obtained a Masters degree in Computer Engineering from Loyola College, and a Bachelor of Science in Computer Engineering from Syracuse University.

Dr. Tae (Tom) Oh is an Associate Professor in the Dept. of Information Sciences and Technology and Dept. of Computing Security at Rochester Institute of Technology (RIT). His research focus has been mobile device security, smart grid, mobile ad-hoc networks and security, and cyber security. He has over 20 years of experience in networking and telecommunication as an engineer and research for several telecom and defense companies. Before he joined RIT, he was a Principle Systems Engineer at Rockwell Collins where he was leading the advanced architecture development of Mobile Ad-hoc Networks for Military Applications. Additionally, he lead/supported OPNET modeling and simulation team for secret projects and actively pursued funding from DARPA, Air Force Research Lab and Army Research Lab. Prior to Rockwell, he worked for Ceterus Networks, Ericsson, Nortel Networks and Raytheon. Dr. Oh received his BS in Electrical Engineering from Texas Tech University in 1991 and received MS and PhD in Electrical Engineering from Southern Methodist University (SMU) in 1995 and 2001, respectively, while working for telecommunication and defense companies. He has published numerous technical articles and holds several patents as well as several teamwork and teaching awards from Nortel and Ericsson. Lastly, he is a Technical Editor of IEEE Communications Magazine and IEEE Networks, and received several grants from Office of Naval Research, Rochester General Hospital, and mobile security companies.

Pratik C. Patel is presently pursuing his M.Tech. in Computer Science & Engg. (Cyber Security) in the Dept. of Computer Engineering at Defence Institute of Advanced Technology, Pune, India. His research interests include Data Mining, Digital Forensics, Soft Computing. His

current research is focused in the area of Data Theft Detection.

Dhananjay S. Phatak earned a PhD degree in computer engineering at the University of Massachusetts, Amherst (UMASS) in 1994, the MSEE degree in microwave engineering at UMASS in 1990, and the B. Tech degree in Electrical Engineering from the Indian Institute of Technology (IIT), Bombay. Dr. Phatak has been a member of the technical program committee of the IEEE Biannual symposium on Computer Arithmetic (IEEE-ARITH) from 1999 through 2009. He also served a three-year term as an Associate Editor of the IEEE Transactions on Computers from January 2002 through December 2005. He was a recipient of the National Science Foundation's (NSF) Career award in FY'99. Currently, he is an associate professor of computer science at the University of Maryland, Baltimore County (UMBC) in the CSEE Dept., and a member of UMBC's Center for Cybersecurity as well as the Cyber Defense Laboratory. Currently, his main research interests include computer and network security, number theory, computer arithmetic algorithms and their VLSI realizations. For further information, please see http://www.csee.umbc.edu/~phatak/

H. Raghav Rao has a Ph.D from Purdue University, an M.B.A from Delhi University, and a B.Tech. from the Indian Institute of Technology. His interests are in the areas of management information systems, decision support systems, and expert systems and information assurance. He has chaired sessions at international conferences and presented numerous papers. He has authored or co-authored more than 100 technical papers, of which more than 60 are published in archival journals. His work has received best paper and best paper runner up awards at AMCIS and ICIS. Dr. Rao has received funding for his research from the National Science Foundation, the Department of Defense and the Canadian Embassy and he has received the University's prestigious Teaching Fellowship. He has also received the Fulbright fellowship in 2004. He is a co-editor of a special issue of The Annals of Operations Research, the Communications of ACM, associate editor of Decision Support Systems, Information Systems Research, and IEEE Transactions in Systems, Man and Cybernetics, and co-Editor- in -Chief of Information Systems Frontiers.

Leonid (Leon) Reznik is a Professor of Computer Science at the Rochester Institute of Technology, New York, USA. He received his BS/MS degree in Computer Control Systems in 1978 and a Ph.D. degree in 1983 and has worked within the industry and academia in Russia, Australia and USA since 2002. Prof. Reznik is an author of the textbook Fuzzy Controllers (Elsevier-Butterworth-Heinemann, Oxford, 1997) and an editor of Fuzzy System Design: Social and Engineering Applications (Physica Verlag, 1998), Soft Computing in Measurement and Information Acquisition (Springer, 2003), Advancing Computing and Information Sciences (Cary Graphic Arts Press, 2005). Dr. Reznik's research concentrates on study and development of intelligent computing systems for control, sensor networks and systems as well as cyber security applications.

Billy Rios is currently the director of consulting at Cylance and is the Chair of the Operational Security Testing panel at the NBISE. Previous to this, he was a Team Lead for Google where he studied emerging security threats and technologies. Billy was one of the primary security engineers for Google Plus, the new social network by Google. Before Google, Billy was a Security Program Manager at Microsoft where he helped secure several high profile software projects including Internet Explorer and Microsoft Online. Prior to his roles at Google and Microsoft, Billy was a penetration tester for various consulting firms. Before his life as a penetration tester, Billy worked as an Information Assurance Analyst for the Defense Information Systems Agency (DISA). While at DISA, Billy helped protect Department of Defense (DoD) information systems by performing network intrusion detection, vulnerability

analysis, and incident handling, Before attacking and defending information systems, Billy was an active duty Officer in the United States Marine Corps where he served as an OIC, Platoon Commander, and Company Executive Officer. Billy is an accomplished public speaker and published author. He has authored and contributed to several books, most notability: "Hacking: The Next Generation" and "Inside Cyber Warfare: Mapping the Cyber Underworld", both published by O'Reilly Media. Billy has also presented at such prestigious security conferences as Black Hat, RSA, NATO CCDCOE, Microsoft's Blue Hat, DEFCON, ToorCon Seattle, and HITB Security conference. Billy is cited in numerous security advisories for research on attacking Industrial Control Systems, URI and protocol handlers, content ownership issues (such as the GIFAR attack), DNS rebinding attacks (against Flash and the Java Virtual Machine), and was previously credited for discovering vulnerabilities in Microsoft Windows and Adobe PDF Reader.

Jim Samuel is an experienced global business strategy management consultant who has worked on diverse projects in Asia, Europe and the Americas. Jim has worked with various global financial corporations including Citibank and ABN AMRO, coupled with an established interest in academia, research, teaching and corporate training. He demonstrates a powerful mix of solid financial expertise, thought leadership, strategic thinking and creative ideation. Domains of research and professional interest include technology, capital markets, risk, integrative dynamics and strategic integration, information economics, information systems and market microstructure. Over the past 22 years, Jim has been involved in the development of numerous successful companies and projects internationally. Jim is a much sought after speaker who has developed strong academic credentials. He is presently completing his doctoral dissertation at the City University of New York. Armed with advanced business degrees, he taught at various business schools internationally, including the prestigious Zicklin School of Business in New York City, and conducting corporate education sessions for executives from Verizon, Pfizer and Johnson & Johnson, amongst others.

Dr. G.K. Shukla is a Professor at the Indian Institute of Management Lucknow. He has a M.Sc. in Statistics from Lucknow University and a Ph.D. in Statistics from Edinburgh in the U.K. He serves on the Advisory Board and Committees of various journals and conferences including: the UPES Journal of Energy & Innovation and the Annual Conference of Society of Statistics, Computer, and Applications.

Victor Skormin is a Professor of Electrical and Computer Engineering at Binghamton University (State University of New York - SUNY), Binghamton NY. He has a MS (1968) degree from the Kazakh Polytechnic Institute, Alma-Ata, U.S.S.R, and a Ph.D. (1974) degree from the Institute of Steel and Alloys, Moscow, U.S.S.R.. While at Binghamton, he established a control engineering curriculum, a laboratory for laser communication research, pioneered computer network security research at Binghamton, and established and directed the Center for Advanced Information Technologies. He received the SUNY Chancellor's Awards for Excellence in Teaching and for Excellence in Research, the IEEE Award "For Leadership in Establishing University Industry Links...", and the rank of Distinguished Service Professor. Dr. Skormin's research in the areas of technical diagnostics, laser communications, and computer network security has been supported by the National Science Foundation, NASA and the Air Force. He served as a consultant to Eastman Kodak, General Electric, Corning Glass Works, Martin Marietta, Universal Instruments and the Air Force Research Laboratory. Twice in his career Dr. Skormin has been appointed by the National Research Council as an Air Force Senior Research Associate. Since 2001 he is an organizer of the on-going bi-annual International conference "Mathematical Methods, Models and Architectures for Computer Networks Security" in St. Petersburg, Russia (sponsored by the US Air Force and Navy).

He is an Honorary Professor of the Kazakh National Technical University, Almaty, Kazakhstan, and an International Member of the Russian Academy of Navigation and Motion Control Sciences. Dr. Skormin is an author/editor of several books and a large number of research papers; he supervised 20 PhD dissertations, and served as an Editor for Space Systems of the IEEE AES Transactions. He is a Senior Member of IEEE.

Kriti Sharma is a graduate student in the Department of Information Sciences and Technologies (IST) at the Rochester Institute of Technology.

Raj Sharman is an Associate Professor in the Management Science and Systems Department of the State University of New York at Buffalo. His expertise is in the areas of Information Assurance and the development of Biologically Inspired Computer Security Models, Disaster Preparedness and Response Management, Patient Safety and Health Care Systems. He has published widely in National and International journals and is the recipient of several grants from university and external agencies, including the National Science Foundation. He received his PhD in Computer Science and a Master of Science degree in Industrial Engineering from the Louisiana State University. He received his Bachelors degree in Engineering and Masters Degree in Management from the Indian Institute of Technology, Bombay, India.

Alan T. Sherman is an associate professor of computer science at the University of Maryland, Baltimore County (UMBC) in the CSEE Dept. and Director of UMBC's Center for Information Security and Assurance. His main research interest is high-security voting systems. He has carried out research in election systems, algorithm design, cryptanalysis, theoretical foundations for cryptography, applications of cryptography, and cloud forensics. Dr. Sherman is also a private consultant performing security analyses, and an editor for Cryptologia. Sherman earned the PhD degree in computer science at MIT in 1987 studying under Ronald L. Rivest. www.csee.umbc.edu/~sherman

Dr. Upasna Singh is presently working as an Assistant Professor in the Dept. of Computer Engineering at Defence Institute of Advanced Technology, Pune, India. Her research interest includes Data Mining, Digital Forensics, Machine Intelligence, Soft Computing. She has been working in these areas from past 7 years.

Andrew Soknacki is currently finishing up his B.C.S. at Carleton University in Ottawa, Canada. His research interests are predominantly composed of Software Defined Radio applications and research.

Anil Somayaji is an Associate Professor in the School of Computer Science at Carleton University in Ottawa, Canada. He received a B.S. (1994) in Mathematics from the Massachusetts Institute of Technology and a Ph.D. (2002) in Computer Science from the University of New Mexico. He has served as the program committee chair of the New Security Paradigms Workshop and has served on the program committees of major computer security venues including ACM CCS, USENIX Security, ACSAC, and RAID, among others. His research interests include computer security, operating systems, complex adaptive systems, and artificial life.

Colin Szost is an undergraduate student at the Rochester Institute of Technology.

William (Bill) Stackpole is an Associate Professor and has been on the faculty at the Rochester Institute of Technology since 2001. He teaches in the Department of Networking Security and Systems Administration primarily in the areas of computer system security, mobile security and forensics.

Wade Trappe is Professor in the Electrical and Computer Engineering Department at Rutgers University, and Associate Director of the Wireless Information Network Laboratory (WINLAB),

where he directs WINLAB's research in wireless security. Professor Trappe has served as an editor for IEEE Transactions on Information Forensics and Security (TIFS), IEEE Signal Processing Magazine (SPM), and IEEE Transactions on Mobile Computing (TMC). He served as the lead guest editor for September 2011 special issue of the Transactions on Information Forensics and Security on "Using the Physical Layer for Securing the Next Generation of Communication Systems" and also served IEEE Signal Processing Society as the SPS representative to the governing board of IEEE TMC.

Rohit Valecha is a candidate for the doctoral degree in Management Science and Systems at the State University of New York, Buffalo, NY. His research interests include Disaster Preparedness and Response Management, Information Assurance and Patient Safety and healthcare Delivery Systems.

Merrill Warkentin is a Professor of MIS and the Richard Puckett Notable Scholar in the College of Business at Mississippi State University. He earned his Ph.D.in MIS at the University of Nebraska - Lincoln. His research focuses on behavioral issues in IS Security and electronic group collaboration. He has authored over 250 manuscripts and six books. He is serving as Associate Editor for MIS Quarterly, European Journal of Information Systems, and Information & Management. He has chaired several international conferences, including IFIP and WISP. His work has been supported by the U.S. Navy, NSA, the IRS, the UN, Homeland Security, IBM, and others. He has been a visiting scholar at over two dozen universities in eight nations, and has served as an ACM National Distinguished Lecturer. His work has appeared in MIS Quarterly, Decision Sciences, European Journal of Information Systems, Decision Support Systems, DATA BASE for Advances in Information Systems, Information Systems Journal, Communications of the Association for Information Systems, Communications of the ACM, and other journals and in numerous books.

INDEX OF AUTHORS