ACADEMIC TRACK
6th Annual Symposium on Information Assurance
>> ASIA '11
New York State Division of Homeland Security and Emergency Services
OCS
Office of Cyber Security
University at Albany
Sapientia et sua et docendi causa
1844
State University of New York
The NYS Forum

conference proceedings

6th Annual Symposium on Information Assurance (ASIA '11)

Symposium Chair:

Sanjay Goel
Information Technology Management, School of Business
University at Albany, State University of New York

Academic Track of 14th Annual NYS Cyber Security Conference
Empire State Plaza Albany, NY, USA
June 7-8, 2011

Proceedings of the 6th Annual Symposium on Information Assurance
Academic track of the 14th Annual 2011 NYS Cyber Security Conference
June 7-8, 2011, New York, USA.

Symposium Chair

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

Submissions Chair

Damira Pon, University at Albany, SUNY

Program Committee

External Reviewers

Note of Thanks
We would like to express our appreciation to all of the sponsors which supported the symposium.

CONFERENCE KILOBYTE SPONSORS

Symantec

McAfee

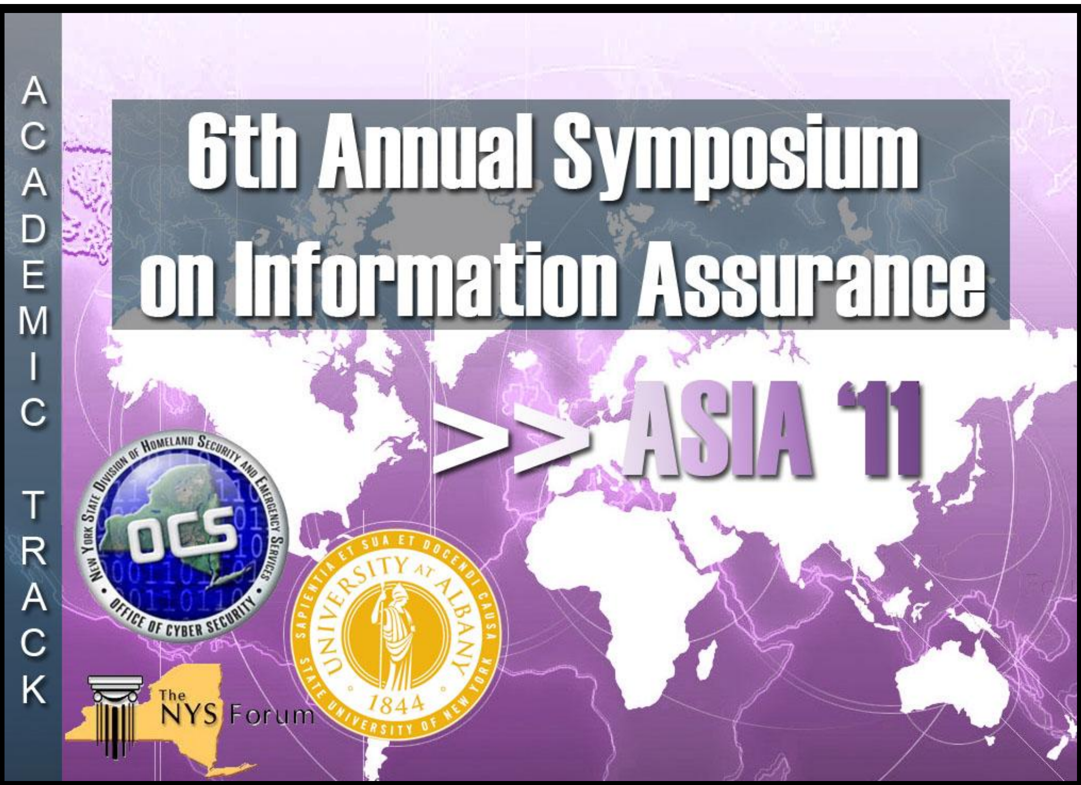
TAG SOLUTIONS
MAXIMIZE YOUR TECHNOLOGY INVESTMENTS

CONFERENCE MEGABYTE SPONSOR

at&t

SYMPOSIUM DINNER SPONSOR

SCHOOL OF BUSINESS
UNIVERSITY AT ALBANY State University of New York


ACADEMIC TRACK
6th Annual Symposium on Information Assurance
>> ASIA '11
New York State Division of Homeland Security and Emergency Services
OCS
Office of Cyber Security
University at Albany
Sapientia et sua et docendi causa
1844
State University of New York
The NYS Forum

conference proceedings

6th Annual Symposium on Information Assurance (ASIA '11)

Symposium Chair:
Sanjay Goel

Information Technology Management, School of Business
University at Albany, State University of New York

Academic Track of 14th Annual NYS Cyber Security Conference
Empire State Plaza Albany, NY, USA
June 7-8, 2011

MESSAGE FROM SYMPOSIUM CHAIR

Welcome to the 6th Annual Symposium on Information Assurance (ASIA'11)! This symposium 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, this symposium has been a great success with excellent papers and participation from academia, industry, and government and highly attended sessions. This year, we again have an excellent set of papers, invited talks, and keynote addresses.

The keynote speaker for ASIA this year is Patricia A. Hoffman, Principal Deputy Assistant Secretary of the United States Department of Energy Office of Electricity Delivery and Energy Reliability. She provides leadership on a national level to enhance security and reliability of energy infrastructure including recovery of this infrastructure from any disruptions. With her expertise, she will provide a unique perspective for how innovations to the electric grid will generate security needs. The symposium has papers in multiple areas of security, including online profiling of cyber crime, user behavior analysis, data privacy issues, forensics, and cyber security education.

I would like to thank the talented program committee that has supported the review process of the symposium. 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 and that each program committee member was not assigned to review more than three papers. The papers and the reviews were also personally read concurring with reviewer assessment. The acceptance rate for this year's conference is 50% with several high-quality papers and invited speakers on the agenda. Our goal is to keep the quality of submissions high as the symposium matures. The program committee serves a critical role in the success of the symposium and we are thankful for the participation of each member. I would also like to thank my multi-talented submissions chair Damira Pon, who not only helps manage the submissions, but also works on preparation of the program book and deals with other logistics.

We were fortunate to have extremely dedicated partners in the New York State Division of Homeland Security and Emergency Services 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 megabyte sponsor AT&T, the symposium dinner sponsor – the University at Albany's School of Business and the conference kilobyte sponsors: Symantec, McAfee, and TAG Solutions 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. Next year, again we are planning on holding the symposium. If you would like to propose a track, 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 2012.

Sanjay Goel

Sanjay Goel, ASIA '11 Chair
Director of Research, CIFA
Associate Professor, School of Business

SYMPOSIUM ON INFORMATION ASSURANCE AGENDA

DAY 1: Tuesday, June 7, 2011 (8:00am – 4:00pm)

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

MORNING SESSION & KEYNOTE – Mtg. Room 6 (9:00am – 10:00am)

Welcome Address: Welcome address: Thomas F. Smith, George Philip, Deborah Buck
Keynote: Roberta Stempfley, *Deputy Assistant Secretary & Director for the National Cyber Security Division*

ALL-CONFERENCE FEATURED SPEAKER – Mtg. Room 6 (10:00 – 11:30am)

What Should Keep You Up at Night: The Big Picture and Emerging Threats
Ted Demopoulos, *SANS Institute*

LUNCH / VISIT EXHIBITORS – Base of the Egg (11:30am – 1:00pm)

ALL-CONFERENCE FEATURED SPEAKER – Mtg. Room 6 (1:00 – 1:45pm)

Succeeding in a Cyber World
Harry D. Ruduege, Jr., *Chairman, Deloitte Center for Cyber Innovation*

BREAK / VISIT EXHIBITORS – Base of the Egg (1:45 – 1:50pm)

SYMPOSIUM SESSION 1: Modeling User/Hacker Behavior (1:50 – 2:50pm)

Chair: Kevin Williams, *Dean of Graduate Studies, University at Albany, SUNY*

Toward Cyber Crime Profiling: Cyber Stalking
Peter R. Stephenson, *Norwich University, Northfield, Vermont*
Richard D. Walter, *Vidocq Society*

Email Behavior Profiling based on Attachment Type and Language
Onur Polatcan, MS., *VA Medical Center, Indianapolis, Indiana*
Sumita Mishra & Yin Pan, *Rochester Institute of Technology, Rochester, New York*

BREAK / VISIT EXHIBITORS – Base of the Egg (2:50 – 3:00pm)

SYMPOSIUM SESSION 2: Sensor Network / User Security (3:00 – 4:00pm)

Chair: *Damira Pon, University at Albany, SUNY*

Enterprise Mobile Security using Wireless Sensor Networks: Extending a Secure Wireless Sensor Network to the Android Smart Phone Platform
Biswajit Panja, Kevin Highley & Priyanka Meharia,
Morehead State University, Morehead, Kentucky

Security of Computer Use Practice: The Case of Ordinary Users Survey
Leon Reznik, Vincent J. Buccigrossi III, Justin Lewis, Asif Dipon, Stefanie Milstead, Nathan LaFontaine, Kenneth Beck & Holden Silvia, *Rochester Institute of Technology, Rochester, New York*

SYMPOSIUM ON INFORMATION ASSURANCE AGENDA, CONT'D.

DAY 2: Wednesday, June 8, 2011 (7:00am – 3:40pm)

REGISTRATION / VISIT EXHIBITORS – Base of Egg (7:00 – 8:45am)

ASIA Keynote & Best Paper Award – Mtg. Rm. 6 (8:45am – 10:00am)

Introduction: Sanjay Goel, *Symposium Chair*
Welcome Address: Donald Siegel, *Dean, School of Business, University at Albany*

ASIA Keynote: Working to Achieve Cyber Security in the Energy Sector: A Public-Private Partnership Approach
Diane T. Hooie, *Principal Deputy Assistant Secretary, U.S. Department of Energy*

Best Paper Award Presentation

SYMPOSIUM SESSION 3: Data Privacy Protection (10:10 – 11:10am)

Chair: Peter Stephenson, *Norwich University*

Understanding Data Leak Prevention
Preeti Raman, Hilmi Güneş Kayacık & Anil Somayaji, *Carleton University, Ottawa, Canada*

Breaching & Protecting an Anonymizing Network System
Jason W. Clark & Angelos Stavrou, *George Mason University, Fairfax, Virginia*

BREAK / VISIT THE EXHIBITORS – Base of the Egg (11:10 – 11:30am)

SYMPOSIUM SESSION 4: Impact of Breaches / Forensics (11:30am – 12:30pm)

Chair: Anil Somayaji, *Carleton University*

Framing Effects of Crisis Response: Communications on Market Valuation of Breached Firms
Manish Gupta, Raj Sharman & H.R. Rao
University at Buffalo, The State University of New York, Buffalo, New York

Automatically Bridging the Semantic Gap Using C Interpreter
Hajime Inoue, Frank Adelstein, Matthew Donovan & Stephen Brueckner
Architecture Technology Corporation, Ithaca, New York

LUNCH / VISIT THE EXHIBITORS – Base of the Egg (12:30 – 1:30pm)

ALL-CONFERENCE FEATURED SPEAKER – Mtg. Room 6 (1:30 – 2:30pm)

Cyber Criminals: Who are They? Why are They Successful? How Do We Respond?
Kimberly Kiefer Peretti, *PricewaterhouseCoopers*

SYMPOSIUM SESSION 5: Secure Routing / InfoSec Education (2:40pm – 3:30pm)

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

Protection Profile-based Scenario-centric Taxonomy of Secure Routing Protocols in Ad-hoc Networks
Mohammad Iftekhar Husain & Ramalingam Sridhar
University at Buffalo, The State University of New York, Buffalo, New York

A Holistic Modular Approach to Infuse Cybersecurity into Undergraduate Computing Degree Programs
Trudy Howles, Carol Romanowski, Sumita Mishra, & Rajendra K. Raj
Rochester Institute of Technology, Rochester, New York

CLOSING REMARKS (3:30 – 3:40pm)

Sanjay Goel, *Symposium Chair*

TABLE OF CONTENTS

Session 1: Modeling User / Hacker Behavior

Session 2: Sensor Network / User Security

Symposium Keynote Address

Session 3: Data Privacy Protection

Session 4: Impact of Breaches / Forensics

Session 5: Secure Routing / InfoSec Education

Toward Cyber Crime Assessment: Cyberstalking

Peter R. Stephenson, and Richard D. Walter

***Abstract*—The concept of offender profiling in computer-related crime is in its infancy. Virtually no research exists that relates offender profiling explicitly to cyber crimes or cyber criminals. Yet it cannot be denied that, given the breadth of potential suspects in a cyber event, some method of reducing that number to a manageable level followed by the ability to identify a small number of credible suspects is very desirable. Today, much cyber crime is treated by the criminal justice system as special cases of physical crime. There is little argument, however, that there are aspects of computer-related crimes and the criminals who perpetrate them that are unique to the virtual, rather than the physical world. The research described in this paper seeks to establish criteria for analyzing cyber crimes and criminals in the clear, unambiguous context of the virtual world. The authors have hypothesized four general classes of computer-related crime: 1) theft, 2) system attack, 3) personal and 4) terrorism. This paper discusses a specific aspect of the personal class of cyber crime: cyber stalking. The four sub-types discussed are the result of decades of empirical application comprising thousands of cases in the physical world. They have proven reliable in investigation of violent crimes such as rape and murder. An underlying purpose in the current research is application of the sub-types in cyber investigation. This is a "research-in-progress" paper, presenting a hypothesis that will be tested empirically in the next phase of the research. However, we present an exemplar case with which we illustrate the potential use of the profiling techniques presented.**

***Index Terms*—Stalking, Cyber Crime, Harassment, Cyberstalking, Internet, Sub-Types, Power Assertive, Power Reassurance, Anger Retaliatory, Anger Excitation, Typology**

I. Introduction

THE Internet is a universal enabler. It not only provides opportunities for research and commerce heretofore unavailable to most people, it also provides a means for criminal activity potentially unrivaled in the pre-Internet age. Because the Internet provides the illusion – and, sometimes, the reality – of anonymity, those wishing to pursue criminal activity find the Internet a safe and fertile ground for their efforts. One of those activities, enabled by the Internet and the current state of the World Wide Web (sometimes referred to as "web 2.0"), is cyberstalking.

There has been some discussion in the literature of cyberstalking as an extension of physical stalking [2][6][7], however, McFarlane and Bocij [8] disagree. The differentiation of cyberstalking as a unique act, albeit sharing some of the characteristics of physical stalking, is an important point for providing a typology of cyberstalkers that can be used efficaciously by investigators. Our position in this regard supports McFarlane and Bocij, however, we find their typologies limited in regards to their use as an investigative tool.

With little concrete information on stalking in the physical world and even less in the virtual world [5], the notion of developing suitable typologies for cyberstalkers has not been well-developed. McFarlane and Bocij [8] have proposed a cyberstalker typology – vindictive, composed, intimate and collective – but this typology does little to differentiate individual cyberstalkers such that investigators can focus on unique suspects based upon their individual behaviors. It is that focus with which this paper deals.

II. Background and Prior Research

There are several theories from the more familiar crime of physical stalking that can be considered for inclusion in the realm of cyberstalking. One such theory is routine activity theory (RAT) [3][4]. Simply, RAT says that crime is inevitable (motivated offenders) and that if a suitable target is unprotected (absence of a capable guardian), he or she is a potential victim. The analog in the virtual world says that if a target frequents public sites and is not protected, he or she may fall victim to some form of cyber mischief.

RAT in cyberspace is most easily illustrated by the vulnerability of many computer users to malware (malicious software such as viruses) and hacking risks. For example, Internet surfers who frequent pornography sites are more likely to face security risks such as a virus or session hijacking (34.2% of free pornography sites and 11.4% of for-pay sites are affected) than those who do not frequent those sites [9]. Similarly, one may theorize that individuals who frequent social networking sites such as Facebook or use products such as AOL Instant Messenger heavily are placing themselves in a dangerous position relative to cyberstalking.

The key to avoiding compromise under the RAT is protecting oneself. In the case of malware protection, this

Peter R. Stephenson is with the Computer Science and Computer Security and IA Departments of Norwich University, Northfield, VT 05663 USA (e-mail: pstephen@norwich.edu).

Richard D. Walter is one of the co-founders of the Vidocq Society and was a forensic psychologist for the State of Michigan prison system and crime analyst profiler.

consists of avoiding dangerous websites and ensuring that anti-malware protection is installed and up-to-date. In the case of cyberstalking, protection may consist of limiting the amount of personal information the individual makes available online. Holt and Bossler [10] report some success in applying RAT to cyberharassment and cyberstalking.

While RAT offers a good framework for helping potential targets of cyberstalking avoid becoming victims, it does not offer the investigator much assistance in identifying a cyberstalker.

McFarlane and Bocij's typology is limited in that it focuses on broad descriptions of cyberstalkers. These broad descriptions do not offer the granular differentiators that investigators require to conduct a credible, prosecutable cyberstalking investigation. The four types described by these authors place cyberstalkers in groups, but do not differentiate adequately at the individual level – nor are they coupled with an investigative approach that makes them useful to investigators. As clinical descriptions they do, however, have merit if taken in the company of other clinical diagnoses.

As well, the empirical research reported in [8] and [14] are quite useful in understanding the crime of cyberstalking, at least in the UK where much of the research was conducted.

The most promising typology comes from Keppel and Walter [1]. This typology, as it stands, is focused on sexual-related murder. However, we have found that it can be extended cleanly to provide a useful typology for assessing cybercrimes and profiling cyber offenders, in this case, cyberstalkers.

An important distinction must be made between a psychological assessment and a criminological assessment. A psychological assessment focuses upon the clinical aspects (e.g., diagnosis and treatment) of the individual. A criminological assessment focuses upon crime and criminal acts. For the purposes of crime assessment, we examine the criminological – and in this case, the cyber criminological – continuum. The investigator applies the sub-types to the crime and then works outward towards the individual suspects.

The Keppel / Walter Sub-Types

Building off of early research by Groth and Birnbaum [11] and subsequent work by Hazelwood and Burgess reported in an early edition of [12], Keppel and Walter extended the typology of rapists to include rape/murder [1]. The extensibility of this typology, as shown by Hazelwood, Walter, et al, suggests that it is an ideal candidate for examining cyber stalking. It is on these sub-types that we base our research into profiling of cyber crimes and criminals. The profiling of cyberstalkers is a first step in that direction.

The sub-types described in [1] include Power Assertive, Power Reassurance, Anger Retaliatory, and Anger Excitation. Briefly, this paper describes these subtypes in the following section although we are interested primarily in Power Assertive and Power Reassurance when we discuss cyberstalking.

A. Power Assertive

The power assertive (PA) actor is focused upon power and aggression and uses them to control the victim. Forceful intimidation and direct application of force are hallmarks of the power assertive subtype. We extend this into the virtual world by adding, for example, the dimensions of bragging about the actor's stalking accomplishments in such anonymous venues as open forums, social networking sites and anonymous discussion groups, as well as proficiency in computer technology. The PA actor tends to be organized and in the cyber world may be a programmer or fancy him or herself to be a super hacker.

The power assertive actor must maintain his or her authority and does it through increasing the level of arrogance and intimidation that can be observed in emails and other postings. The actor is egocentric and applies his or her ego to maintain dominance. In the PA cyberstalker, the level of control available in the virtual world may not be enough for the stalker to believe that he or she is maintaining control over the victim and, thus, may escalate to a physical meeting in the real world. That meeting can result in rape or rape-murder.

B. Power Reassurance

The power reassurance (PR) actor is similar to the PA actor with some important differences. The big difference is the impact of fantasy on this actor. By fantasy, we mean the difference between reality and what the actor wants and/or believes to be true via magical thinking1. In the organized actor, this may play out as celebrity stalking, as an example, where the actor believes that the celebrity is in love with him or her. In contrast, the disorganized actor may target on either side of his or her age group or, if within the same age range, he or she may focus upon the challenged – physically, mentally, or naive – for the exploration and exploitation of power.

The PR actor needs to reinforce his or her view of him- or her-self and this sometimes presents as an underlying lack of self-confidence and sophistication. This may, though, be part of the PR actor's fantasy. The actor will attempt to engage the victim in his or her fantasy and will increase aggression more moderately than the PA actor. However, when that does not work, the PR behavior may escalate to PA. The PR actor is less organized than the PA and may leave more clues that enable the tracing of the cybertrail more easily.

In the cyber world, the PR actor may use a doctored photo and create a persona that he or she believes will be attractive to the victim. Although the actor may present as being low on self confidence, he or she will attempt to appear confident and when the online connection ceases to satisfy the actor's fantasy, he or she may attempt to escalate to a meeting in the

[1] Magical thinking is a volitional associational thought pattern, without boundaries, that creates a desired result without rational thought. Magical thinking requires only belief and want, rather than fact and examination.

real world.

C. Anger Retaliatory

The anger retaliatory (AR) actor is full of hostility and will act that rage out against the specific source or, if the source is unavailable, a symbolic target that represents the true cause of real or imagined wrongs. While the target of the rage may be one or more persons, the real cause may be one or more persons or an organization.

AR actors in cyberspace do not usually escalate to meetings in the physical world and, in fact, AR behavior is rarer than PA or PR behavior in the online world.

D. Anger Excitation

Anger excitation (AE) actors are sadistic and focus their activities on terrorizing the victim. The level of aggression increases until the actor achieves the destruction of the target. Because AE actions are difficult to achieve in cyberspace, the AE type cyberstalker is very rare.

III. Using the Sub-Types in Cyberspace

The sub-types are applied very specifically in criminal profiling and profiling stalkers in cyberspace is no exception. Simply put, the profiler begins by characterizing the crime based upon the evidence available. The evidence, in this case, includes interviews with victims, forensic analysis of the victim computer(s), Internet Service Provider (ISP) logs, subpoena results from ISPs, social networking sites and other online portals that were involved in accessing the victim. This results in a profile of the crime that the investigator can match with the profiles of suspects.

Since most stalkers in the physical world are known by their victims, we might assume that the same is true in the online world [13]. This turns out to be the case, but the dynamics of that familiarity are somewhat different in cyberspace. In the online world, the balance of former significant others versus new “friends” met online in chat rooms, social networking sites, etc. is tilted towards those met online.

However, Bocij does not agree completely [14]. He reports that “… there is always [his emphasis] some kind of relationship between the offline [nb – physical world] stalker and his victim.” Bocij makes this statement as a differentiator between physical world and cyber world stalkers. He contends that cyberstalkers do not always know their victims.

This does not take into account the extensive use of social networks in cyberspace where interactions can become very personal even though the actors have never met in person. For a PR cyberstalker, such limited contact online can develop into a fantasy that results in aggressive cyberstalking and sometimes, an escalation to a real world physical meeting, often with serious consequences.

Statistically most physical stalkers are men and most victims are women [13]. There is little evidence to dispute that balance in cyberspace, although validating it is one of the goals of the empirical portion of this research.

A. Investigation

Analysis of a cyberstalking incident should begin with a clear understanding of the events making up the incident. That includes detailed interviews with the victim and a detailed forensic analysis of the victim’s computer. PA cyberstalkers are likely to have a moderate to high level of computer skill and that will be evident in anonymization of emails and other messages or direct access, if any, to the victim’s computer. Often the victim will have deleted offensive emails and other postings. Those will need to be recovered forensically from the victim’s computer.

Analysis of the activities of the cyberstalker through reconstruction of communications with the victim is the next step. That likely will require subpoenas ISPs, portal operators, email services, and social networking sites. There is a high likelihood that some form of alias will have been used by the cyberstalker. It is important to cross that alias to a real person. That chain of evidence – also called a cybertrail – may be a many-headed hydra leading in a variety of directions. A single cyberstalker may use multiple aliases.

Once the profile of the crime is complete it must be matched with that of the individual cyberstalker. The cyberstalker’s alias is then traced to a real person and that person is profiled using the same sub-types. That may be done by performing extensive searches on the Internet to find other examples of the suspect’s activities or by analysis of the known characteristics of the identified individual. If the suspect’s profile matches the profile of the event, the final step is to analyze the applicable cybertrail to establish that there was, in fact, contact with the victim.

When cyberstalking extends into the real world, evidence gathered through this process can be of material assistance to investigators. Because one of the main differences between physical and cyberstalking is the impact of geography – physical stalkers must be in the geographic vicinity of their victims, while cyberstalkers do not need to be [13] – an important aspect of cyberstalking-turned-physical is geography.

It is important to note that a PR cyberstalker can escalate to PA, but going the other way is very unlikely. The PR cyberstalker initially may take a somewhat gentler approach towards fulfilling his or her fantasy with the victim than will a PA cyberstalker. When that does not produce results, the cyberstalker may become more aggressive and the characteristics of the PA come to the fore. If a cyberstalker starts as PA and becomes PR, the investigator should be suspicious that he or she is being gamed by the subject. It is likely that the actor is PA.

One more important point is worth mentioning. It is less usual for an actor to present as only one type than to demonstrate some balance of more than one.

For example, a PA cyberstalker may have a bit of AR that tends to present as anger towards the victim. However, the investigator should be alert when developing motive to the dominant type, which in this case is PA. The motive is power and control over the victim. The anger may simply be a manifestation of the cyberstalker's need to control and is more of a tool than a complete typology.

IV. Case Example

In the early 1990s, one of the authors worked on a stalking case where the victim was a woman in the human resources department of a medium-size company. She had been employed previously by another organization and had been forced to fire a man who subsequently stalked her physically for some time. As a result, she left the organization because there seemed to be nothing that the organization would do to protect her and the actor was a very violent man.

Several years had passed when the cyberstalking and harassment (cyber harassment is a superset of cyberstalking for the purposes of this example) began, but the harassing emails showed a detailed knowledge of the earlier events. She naturally assumed that it was the same person.

Upon completing an investigation – which did not include profiling – the actor was found to be a co-employee of the victim. The victim had been hired, in part, to control the behavior of other HR employees, especially in their recruiting practices, and this particular employee resented that control. She, therefore, used cyberstalking to reestablish her control and power within the department.

A. Analysis

This was a classic PA cyberstalking. The actor used email with anonymization to stalk the victim and increased the level of aggression to the point where the victim began to fear for her life and considered leaving the company's employ. This, of course, was the objective of the cyberstalker.

In her daily work, the actor could be seen as PA. She was controlling, a bit of a loose cannon and attempted to intimidate coworkers and supervisors into letting her have her way and exercise her duties as and when she wished.

Matching the obvious PA characteristics with the PA nature of the co-worker would have pointed to the actor immediately, but unfortunately, cyber profiling techniques were not developed then even as they are not now.

There is the obvious argument that there is no guarantee, given the size and dispersal of the online world, that the cyberstalker would be anywhere near the physical proximity to the victim or that there would be a connection that would assist investigators in identifying the suspect.

However, there are numerous tools today that can aid in that identification. At the time of the incident, those tools did not, of course, exist. That being said, the statistical connection between attackers and victims in the physical world may tend to be replicated in cyberspace [13]. If that is the case, as it was in the case example, identifying suspects is practical.

Additionally, tracing the cybertrail of the stalker can help investigators identify the suspect regardless of where he or she might be geographically located relative to the victim.

This case example demonstrates the potential benefits of developing cyber profiling. Establishing parallels between the physical and online worlds is an objective of future phases of this research.

Once the profiles of the incident and the potential suspects had been completed, tracing the cybertrail would have led inevitably to the actor. The actor was in the process of escalating her cyberstalking into the physical world by threatening the victim's teenage son – escalating the level aggression – and placing nails under the son's car tires.

Although the cyberstalker herself did not have a high enough skill level to perform the anonymizing of the stalking emails, her husband did and in true PA fashion, the actor got her husband to create and send the emails for her. Involving someone with greater computer skills through intimidation is not uncommon with PA cyberstalkers.

Future Research

As this paper shows, extending the sub-types to the on-line world is feasible. Next steps in this research include performing empirical research using actual cases, examining parallels between the online and physical worlds, and extending the sub-types to the other classes of cyber crime.

Crime assessment in the physical world includes examining activities during the crime as well as pre-crime and post-crime activities. These are aligned with the sub-types to understand the nature of the crime and then applied to suspects. At this point, investigators develop profiles of the suspects using the sub-types and match them to the crime assessment. Typically this analysis will point to one or more viable suspects. Future research will test this approach in the digital world.

V. Conclusion

The development of a reliable method of crime assessment and offender profiling for cyber crimes is both desirable and, in today's online environment, necessary. Unfortunately, most efforts at this so far have been clinically focused within the psychological domain as opposed to applying the criminological continuum and being intended for the investigator of cyber incidents.

Assessing cyberstalking incidents and profiling cyberstalkers using the Keppel/Walter sub-types is an excellent place to start developing this investigative capability because there is a close correlation between physical stalking and cyberstalking. The authors hypothesize, however, that the sub-types can be extended to all forms of cyber crime: theft, system attack, personal, and terrorism.

Acknowledgment

Peter Stephenson would like to acknowledge the significant assistance and support from Richard Walter. Mr. Walter's tutelage on the sub-types and co-authoring of this paper were – and are – invaluable. Also, the resources of the Norwich University School of Business and Management Department of Computing, and, particularly, Dean Frank Vanecek and Professor William Jolley have been valuable in supporting this research. Finally, the Norwich University Center for Advanced Computing and Digital Forensics has played and

will continue to play an important part in developing the concept of cyber profiling into a useful investigative tool.

REFERENCES

[1] R. D. Keppel, and R. Walter, "Profiling killers: a revised classification model for understanding sexual murder," *International Journal of Offender Therapy and Comparitive Criminology*, vol. 43, no. 4, pp. 417-437, 1999.

[2] M. L. Pittaro, "Cyber stalking: an analysis of online harassment and intimidation," *International Journal of Cyber Criminology*, vol. 1, no. 2, pp. 180-197, 2007.

[3] L. E. Cohen, and M. Felson, "Social change and crime rate trends: a routine activity approach," *American Sociological Review*, vol. 44, no. 4, pp. 588-608, 1979.

[4] E. E. Mustain, and R. Tewksbury, "A routine activity theory explanation for women's stalking victimization,". *Violence Against Women*, vol. 5, no. 1, pp. 43-62, 1999.

[5] L. McFarlane, and P. Bocij, "Cyber stalking: defining the invasion of cyberspace," *Forensic Update*, vol. 1, no. 72, pp. 18-22, 2003.

[6] E. Ogilvie, "Cyberstalking," in *Trends and Issues in Crime and Criminal Justice,* no. 166. Canberra, Australia: Australian Institute of Criminology, 2000, pp. 1-6.

[7] A. W. Burgess, and T. Baker, "Cyberstalking," in *Stalking and psychosexual obsession:Psychological perspectives for prevention, policing and treatment*, J. Boon, and L. Sheridan, Eds. Chichester, UK: Wiley, 2002, ch. 12.

[8] L. McFarlane, and P. Bocij. (2003, September). An exploration of predatory behaviour in cyberspace: Towards a typology of cyberstalkers. *First Monday* [Online]. 8(9). [Cited: January 3, 2011] Available: http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1076/996.

[9] G. Wondracek, C. Platzer, E. Kirda, and C. Kruegel. "Is the Internet for porn? An insight into the online adult industry," in *Proc. 9th Workshop on the Economics of Information Security*, Harvard University, Cambridge, Massachusetts, USA, 2010, pp. 1-14.

[10] T. J. Holt, and A. M. Bossler, "Examining the applicability of lifestyle-routine activities theory for cybercrime victimization," *Deviant Behavior*, vol. 30, no. 1, pp. 1-25, 2009.

[11] N. A. Groth, and H. J. Birnbaum, *Men Who Rape: The Psychology of the Offender.* New York, NY: Plenum Press, 1979.

[12] R. R. Hazelwood, and A. W. Burgess, *Practical Aspects of Rape Investigation (*4th ed*.).* Boca Raton, FL : CRC Press, 2009.

[13] United States. Attorney General to the Vice President. *1999 Report on Cyberstalking: A New Challenge for Law Enforcement and Industry.* Washington DC : United States Department of Justice, 1999.

[14] P. Bocij, *Cyberstalking - Harassment in the Internet Age and How to Protect Your Family.* Westport, CT: Praeger Publishers, 2004.

E-mail Behavior Profiling based on Attachment Type and Language

Onur Polatcan, Sumita Mishra, and Yin Pan

***Abstract*—Protection of confidential information from insider threat is crucial for any organization. In particular, compromise of information via email is relatively easy and can go undetected. We have developed the *Invisible Witness* tool for prevention and detection of information compromise via email. *Invisible Witness* can automatically detect certain patterns across user accounts that indicate covert or malicious activities. Furthermore, this application assists the network administrator with targeted investigation. Many existing applications look for specific text or attachments, but to the best of our knowledge, *Invisible Witness* is the only application that is capable of creating user profiling. Our preliminary results indicate that this application works with over 95% accuracy. Hence, this tool can be used to minimize business risk by helping protect sensitive information from compromise.**

***Index Terms*—Email Behavior, Profiling, Information Security, Email Security**

I. Introduction

EMPLOYEES of any company or organization are privy to many different types of data in their day-to-day operations. The level of sensitivity of data varies and is highly dependent on the nature of business, job profile, etc. [1]. For instance, for a manufacturing company, sensitive information may include customer lists, designs and drawings, budget details, marketing strategies, and so forth. The job of a network/security administrator includes protection of sensitive data and access control.

In this era of globalization, almost every organization includes employees that are bilingual or trilingual; this potentially can be a means of leakage of information. Meanwhile, many of us wear a mask to disguise the “dark side” of our personalities [2]. A person can easily disguise himself by having a beard one day and being clean shaven on another day. One day, a person may use a British accent for a specific purpose and just as quickly change to an American accent.

Consider the role of a salesperson and how he or she may dress to impress a potential client. The salesperson may use verbal and non-verbal communication techniques to reduce any resistance a potential client may have in the persuasive role of salesmanship. Since it is very difficult to gauge the motives of all the employees in any organization by their appearance, it is important to prevent leakage of confidential data by e-mail with content filtering that includes text and attachments.

When used appropriately, an e-mail behavioral profile can dramatically increase the odds of highlighting possible malicious activity. It can provide the network administrator with a specific time frame that identifies a sender’s activity, and the application promotes a developmental plan that tracks activity and defines the next critical step. Ultimately, it can help organizations to pinpoint their most critical security needs and help IT and upper management better allocate their protected resources.

The rest of this paper is organized as follows. In Section II, the authors introduce the related work in email behavior profiling. Our methodology of identifying abnormal email behaviors based on a user’s profile is presented in Section III. In Section IV, the authors evaluate the proposed approach. Section V addresses the challenges and limitations of the proposed methodology.

This paper is concluded in Section VI with future work in Section VII.

II. Motivation

The motivation and inspiration for this research originated from works of Hadjidj et al. [3]. Their work mainly focuses on keyword searching to authorship attribution of anonymous e-mails and generates reports on e-mail archives. Our tool is significantly different from their work since it works on a live system and it includes a notification tool that would alarm a network administrator when adverse circumstances may occur in an organizational workplace. In our work, we investigate the possibilities of integrating a specialized tool into an e-mail client that would flag suspicious e-mails and attachment violations.

Li, et al. [4] used e-mail filtering, which is based on irrelevant e-mail features and text classification. Their work adopted the algorithm of naïve Bayesian e-mail classification, but our work did not use this algorithm at all.

Because most companies have bilingual or trilingual employees [5], our motivation is to provide an internal tool that will monitor language modifications from English to

O. Polatcan is with the Dep. Asst. Sec for Info. And Technology, VA Medical Center, Indianapolis, IN 46222 USA.

S. Mishra* and Y. Pan are with the Networking Security and Systems Administration, Rochester Institute of Technology, Rochester, NY USA (e-mail*: mishra@rmu.edu).

other languages. For instance, an angry employee who is fluent in French could easily send protected company data using French or any other language to reduce detection within the system.

Nearly 90 percent of data compromises commonly happen by using email as the method of communication [6], [7]. The *Invisible Witness* tool will be very helpful in protecting any organization from internal and external threats where Internet access is unrestricted.

III. Methodology

The *Invisible Witness* tool is designed to monitor abnormal activities based on user profiles. In order to statistically measure the effectiveness of this concept, the authors developed an application that captures the attachment size that users send within a given time period, identifies the language used in the email message, and flags the emails that use non-English languages. The application is designed to work for email servers such as Gmail, Exchange, Lotus Domino, etc. To simulate a real-time scenario, the Gmail server is used for this paper because it would give anyone an opportunity to replicate the same environment without environment setup overhead. For instance, if a user sends an email with an attachment size exceeding the predefined attachment size defined in a user's profile, or the email body was detected using a non-English language, the application would notify the network administrator and profile the message information such as the sender, receivers, attachments size, attachment name, and sent date.

Figure 1 depicts the problem domain; an employee sends an email using an email server such as Gnail, Exchange, Lotus Domino, and so forth. *Invisible Witness* runs in the background and monitors the email server for any activity. As soon as the *Invisible Witness* tool gets the notification from an email server that an email has been sent, it checks the size of its attachment and spell checks the message body for English language. The information gathered is stored in a database.

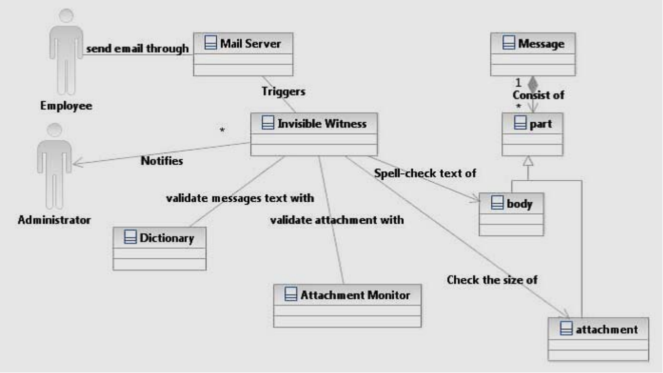


Fig. 1. Unified Modeling Language domain.

A. Use Case Scenario

The use case scenario for *Invisible Witness* when a new mail message is sent is shown in Figure 2 and is outlined below:

Actors: Mail Servers, System, Administrator, Database
Description: System is triggered when a new email has been sent
Trigger: Mail server received a message

Normal Flow:
1. Mail server triggers the system when a new email has been sent
2. System receives an email from the mail server to analyze it
3. System gets a word and passes it through a dictionary
4. Dictionary returns true for correct words
5. Flow repeated from step 3 for each new message sent
6. Percentage of misspelled word is under 25%
7. Message does not have any attachment
8. Exit system

Alternative Flow:
6.1 Percentage of misspelled words are greater than 25%
6.1.1 System connects to mail server
6.1.2 Mail server acknowledges connection
6.1.3 System prepares and sends a message to the administrator
6.2 Message does not have a body
6.2.1 Flow continues from Normal Flow step 7
7.1 Message has attachment
7.1.1 System requests user profile from database
7.1.2 Database calculates and returns the average size of attachment for the user
7.1.3 System compares the attachment size with the average returned from the database
7.1.4 Attachment size is less than the average
7.1.5 Flow continues from Normal Flow step 7
7.1.4.1 Attachment size is greater than average
7.1.4.1.1 System prepares and sends notification to the administrator

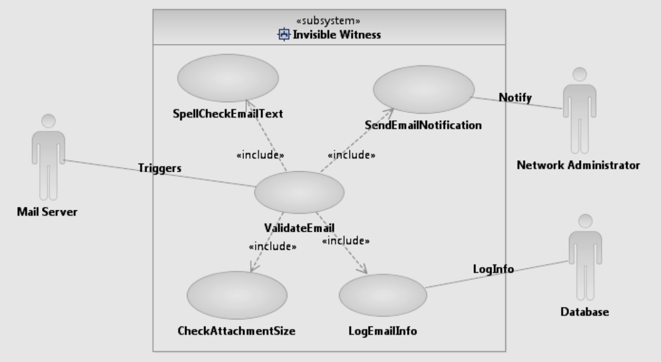


Fig. 2. Use case diagram.

B. Application Methods

Figure 3 shows the Class diagram of the application. The list of classes used in this application and their functionalities is organized as follows:

MailFilter

This is the main class from which an application is managed. It creates an object from both MailDictionary and ConnectionManager to perform the required tasks. Following is a brief description of its methods:

startMonitoring(): will connect to the mail server to retrieve the messages that the system has never yet read. Then, it decomposes the message to its original parts to call one of the methods handleMultipart() or handlePart().

handleMultipart(): will be called in case the message has more than one part to be analyzed. It iterates through all the parts and calls handlePart() for each part.

handlePart(): will search for the parts with plain/text (message body) and gets the content of that part to be validated. It will use isCorrectWord() from the Dictionary class to validate and count misspelled words. The method will also check whether the part is an attachment. If it is, it will get the average attachment size to compare the current attachment size and log the message if the current attachment size is larger than the average.

notifyAdmin(): will notify the administrator if the message exceeds the percentage of misspelled words or if its attachment size exceeds the average.

ConnectionManager

Contains all the functionalities to communicate with the database. It consists of the following methods:

Log(): when an attachment exceeds the average, *Invisible Witness* will use this message to log the attachment information to the database. The method accepts the message itself and the attachment to get enough information to be logged. For instance, the message is needed to get general information such as sender, receiver, and sent date. The attachment is needed to get more detailed information about the attachment itself, such as the attachment size, name, and type.

getLastDaysAttachmentAvg(): *Invisible Witness* will get the average of the last logged activity by the user who sent a message containing at least one attachment to validate the attachment. This method will accept two parameters: 1) sender name, and 2) time period. The method will use sender name to retrieve information about the user and time period in order to calculate the average of the attachments' size logged during this period. In case the method did not find any activity during the specified period, it will call getAltAvg().

getAltAvg(): will calculate the average attachment size using the other period specified in the configuration file, which is usually more than the first period. If no activities were found during this period, it will call getLastAtchSize().

getLstAtchSize(): will retrieve the size of the last attachment that the specified user sent. In case no profile was found, the method will return 0.

getOverAllAvg(): will get the average of all the attachments that the specified user has sent regardless of the attachment size restriction.

removeAliases(): removes the aliases from email addresses. Unlike email addresses, aliases might be changed by the user at any time. So, it is important to remove theses aliases to improve the accuracy of the program.

MailDictionary

Contains all the dictionary-related concepts, such as word validation, dictionary loading, and file specification. It consists of the following methods:

MailDictionary(): accepts the dictionary file name to make it easier to change the diction the application uses to validate the message content.

getWordList(): loads all the dictionary words to make them ready. Loading the words in the data structure is much more efficient than iterating a file each time a word is validated.

isCorrectWord(): validates whether the passed word already exists as a parameter in the dictionary. If it finds the word in the dictionary, it returns true. It returns false otherwise. *Invisible Witness* will basically use isCorrectWord() to validate each word in the body and calculate the average of the misspelled words.

IV. Experimental Evaluation

In this section, the results of the experimental evaluation of *Invisible Witness* are presented. The accuracy of the application was tested for a total of five weeks. However, to make it as easy as possible for the reader, only the last two weeks of data is shown here. Note that the application is not judged by the execution time or limited by the number of sent messages since they are not real-time scenarios. The execution time is not a real-time scenario since the application analyzes any sent message as soon as it is sent. For this reason, the application would not have to analyze a large number of messages at once.

A. *Datasets*

We used our own datasets for the experiment; however, this tool can be applied to the Enron e-mail corpus and other datasets as well [5].

B. *Test Environment*

- Windows 7 Professional 64 bit
- Microsoft Access 2010
- MySQL 5.5
- Gmail server
- Java Mail Library

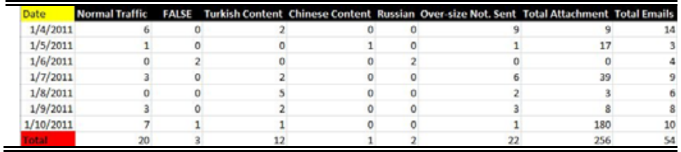
C. *Settings*

It must be noted that when setting up a database, one must ensure that enough characters be given for each column, and the administrator should always read the logs if any error message is generated.

D. *Results*

Invisible Witness 1.0 captures the attachments that are over the average user profile limit and stores them in the database. In our latest version of the tool (*Invisible Witness* 2.0), the application stores all the attachments to a database regardless of the attachment size. Another column in the database called "oversize" was created. If the attachment is larger than its oversize limit, the database displays a “1,” but if the attachment is within parameters, the database will display a “0.” Each time the network administrator receives an email, there is an informational message line that displays "User up-to-date attachment size," which means that the average attachment size is displayed for the specified user.

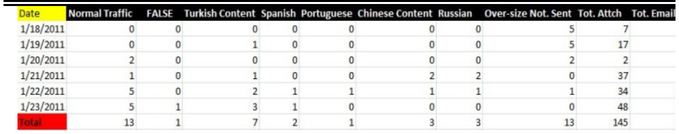
TABLE I
INVISIBLE WITNESS 1.0 RESULTS



Date	Normal Traffic	FALSE	Turkish Content	Chinese Content	Russian	Over-size Not. Sent	Total Attachment	Total Emails
1/4/2011	6	0	2	0	0	9	9	14
1/5/2011	1	0	0	1	0	1	17	3
1/6/2011	0	2	0	0	2	0	0	4
1/7/2011	3	0	2	0	0	6	39	9
1/8/2011	0	0	5	0	0	2	3	6
1/9/2011	3	0	2	0	0	3	8	8
1/10/2011	7	1	1	0	0	1	180	10
Total	20	3	12	1	2	22	256	54

As shown in Table I, *Invisible Witness 1.0* was evaluated over a period of 7 days from January 4-10, 2011. In this test case, a total of 54 e-mails were analyzed. It appeared that 20 of these emails had not been flagged: 3 of them were false positives; 12 of them had Turkish words; 1 email contained Chinese words; 2 emails contained pure Russian; and 22 notifications were sent because of profile attachment size and were logged to the database. Out of 54 emails, a total of 22 emails had attachments totaling 256 attachments, but only 22 of them were over the profile size.

TABLE II
INVISIBLE WITNESS 2.0 RESULTS



Date	Normal Traffic	FALSE	Turkish Content	Spanish	Portuguese	Chinese Content	Russian	Over-size Not. Sent	Tot. Attch	Tot. Email
1/18/2011	0	0	0	0	0	0	0	5	7	
1/19/2011	0	0	1	0	0	0	0	5	17	
1/20/2011	2	0	0	0	0	0	0	2	2	
1/21/2011	1	0	1	0	0	2	2	0	37	
1/22/2011	5	0	2	1	1	1	1	1	34	
1/23/2011	5	1	3	1	0	0	0	0	48	
Total	13	1	7	2	1	3	3	13	145	

As shown in Table II, from January 18-23, 2011, a total of six days of data with 38 emails had been recorded for analysis using *Invisible Witness 2.0.* It appeared that 13 of these emails had not been flagged: 1 was a false positive; 7 of them had Turkish words; 3 emails contained Chinese words; 3 emails contained pure Russian; 1 email was pure Portuguese; 2 emails were Spanish; and 13 notifications were sent because of profile attachment size, were logged to the database, flagging 1 on the “oversize” column. Out of 38 emails, a total of 21 emails had attachments totaling 145 attachments, but only 13 of them were over the profile size.

V. CHALLENGES AND LIMITATIONS

One of the biggest challenges was with Gmail’s IMAP. Gmail doesn't handle all of the standard IMAP flags, such as "\Recent". It is normal behavior when an employee types an email, but it is automatically flagged as “read” by the Gmail server. Since the “Recent” feature didn’t work, we used a “seen” flag to get the data and marked the read message as “unseen” to check only the new messages that the system had not yet read. Even though five weeks of data has been used with over 95 percent accuracy, further testing should be done to improve effectiveness. The dictionary that was used in this test bed has over 237,000 words in it. When an employee uses a word that is not in the dictionary, the system flags it, and for this reason, when the word is not included, the administrator should add new words to get better results.

VI. CONCLUSION

The authors present a robust and scalable application, *Invisible Witness*, for filtering suspicious data from sent emails. This application can capture differing distributions of emails in users’ sent boxes for one or more users, if needed. Over the years, most companies have conducted and will continue to conduct a significant portion of their business through email. Some of this information transmitted by email includes confidential data that if not properly secured, could damage the company’s security severely [8][9]. The significant risks that all companies face are when employees within the organization send confidential information through unprotected email [10]. We hope that organizations will be able to use the *Invisible Witness* tool to minimize their business risk to help protect valuable company information.

VII. FUTURE WORK

We believe this application can be advanced and used as commercial software in the near future. There are four modifications that could be applied:

A. Pushing vs. Pulling of Messages by the Mail Server

The application should be triggered by the mail server whenever a message has been sent. Currently, the application pulls messages from the mail server. The computer resources usage will be limited to when the application is triggered.

B. Graphical Data Representation

The system could be expanded to display graphically the users' profiles as a pie or bar chart to depict what is already logged.

C. Languages Add-On

The system could be expanded to add more language dictionaries so that it would identify the text based on the available language that is loaded on the system.

D. Attachment Analyzer

The system could be expanded to analyze the attachments and determine what type of extensions they have or could be iterated through the hexadecimal object file format.

REFERENCES

[1] D. Ayers, “A second generation computer forensic analysis system,” *Digital Investigation*, vol. 6, pp. S34-S87, 2009.

[2] J. M. Moore, “Your e-mail trail: Where ethics meets forensics,” *Business and Society Review*, vol. 114, pp. 273-293, 2009.

[3] R. Hadjidj, M. Debbabi, H. Lounis, F. Iqbal, A. Szporer, and D . Benredjem, “Towards an integrated e- mail forensic analysis framework,” *Digital Investigation,* vol. 5, no. 3-4, pp. 124-137, 2009.

[4] X. Li, J. Luo, and M . Yin, "E-mail filtering based on analysis of structural features and text classification," in Proceedings of the 2nd International Workshop on Intelligent Systems and Applications (ISA), Wuhan, China, May 2010, pp. 144-147.

[5] C. Forsloff, (2010, September 19). *Behavioral Profiling: Learn to Tell the Truth From Lies* [Online]. Available: http://www.articlesbase.com/psychology- articles/behavioral-profiling-learn-to-tell-the-truth-from- lies-569677.html

[6] S. Garfinkel, P. Farrell, V. Roussev, and G. Dinolt, "Bringing science to digital forensics with standardized forensic corpora," *Digital Investigation*, vol. 6, no. 2009, pp. S2- S11, 2009.

[7] S. Naksomboon, C. Charnsripinyo, and N. Wattanapongsakorn, "Considering behavior of sender in spam mail," in Proceedings of the 6th International Confernece on Networked Computing (INC), Gyeongju, South Korea, May 2010, pp. 1-5.

[8] M. Rogers, "The role of criminal profiling in the computer forensics process," *Computers & Security,* vol. 22, no. 4, 292-298, January 2003.

[9] S. J. Stolfo, S. Hershkop, K. Wang, O. Nimeskern, and C-W. Hu, "Behavior profiling of email," *Lecture Notes in Computer Science,* vol. 2665, pp. 74-90, 2003.

[10] W. N. Gansterer, A. G. K. Janecek, and P. Lechner, "A reliable component-based architecture for e-mail filtering," in Proceedings of the 2nd International Conference on Availability, Reliability, and Security (ARES'07), Vienna, Austria, April 2007, pp. 43-52.

Enterprise Mobile Security using Wireless Sensor Network: Extending a Secure Wireless Sensor Network to the Android Smart Phone Platform

Biswajit Panja, Kevin Highley, and Priyanka Meharia

***Abstract*—Use of wireless sensor technology has increased rapidly in recent years, with many varied applications. In the same timeframe the ubiquitous cell phone has evolved from a simple voice communication device; smart phone technology has enabled consumers to have mobile computing power that rivals the capabilities of desktop computers in the not too distant past. When one takes into consideration the widespread availability of internet connectivity that cell phone networks provide, juxtaposition of wireless sensor technology and cell phone technology is a natural progression. This paper consists of designing and implementing a novel Android application (NodeDroid) that integrates with our existing secure wireless sensor network and addresses the issues encountered in joining two disparate software systems.**

***Index Terms*—Enterprise Mobile Security, Wireless Sensor Networks, Smart Phone Networks, Android OS**

I. Introduction

In recent years, several new uses of wireless sensor networks (WSN) are being developed. Some examples of such applications on WSN include factory automation applications such as asset tracking, machinery condition monitoring, and distributed equipment control. Deployment of WSN includes distribution of thousands of sensors over strategic locations in a structure such as an office building, so that the asset can be constantly monitored in real time both internally and externally. WSN are usually unattended and they also need to be fault-tolerant so that the maintenance cost of the network can be minimized. With the advancement in technology, the sensors today are extremely small, low cost and low powered devices. The low cost of these sensors makes it possible to monitor a large area, reliably and accurately with a network of hundreds or thousands of these wireless sensors.

The data collected by the sensor includes location, so the WSN can be made to be location aware, whenever needed.

In the recent years, owing to technological sophistication, we see a growing trend in enterprise mobility, but the use of mobile phones and other mobile handheld devices have also advanced personal computer capabilities such as supporting email and web access. Wireless sensors come in many configurations and sizes. The data collected by sensors depend upon the type of sensor and the specific sensor network application. A network implementation of these sensors would be appropriate in any situation where there is a need for tracking localized business or weather data. An example is use on airport runways, which is one application envisioned for the software created in this project.

The sensors used in this project form an ad-hoc communications network when powered on. Packet transmissions hop from node to node. In order to minimize power consumption in transmission, thereby maximizing battery life, each node establishes its communication path through a nearby neighboring node. As a result of this communication structure, nodes near the base station forward many packets from outlying nodes. If a sensor's battery is drained, or it otherwise fails, the network is automatically reconfigured around the dormant node. Due to the nature of this communication method, the logical topology of a network region (such an airport runway) won't necessarily match the actual communication paths. Generally, the communication paths will form a rough star pattern as spokes that radiate from the base station (if centrally located).

So far it is assumed that displaying the sensor data information in a base station computer is sufficient [1][2][8]. With the sophistication of smart phones, it is convenient for the users to get processed sensor data in a smart phone. For example: if the base station is collecting data from the sensors deployed in an airport runway to check if there is any possibility of ice formation. Though the alert is generated in the base station, it should also be transmitted to a smart phone like Google Android or Apple iPhone. Also, for business data, it is wise to send information to a smart phone for immediate action.

This project is supported by Kentucky Science and Technology Corporation.

B. Panja* and K. Highley are with Morehead State University, Morehead, KY 40351 USA (e-mail*: b.panja@morehead-st.edu).

P. Meharia is with the University of Kentucky, Lexington, KY 40506 USA (email: pmema2@uky.edu).

In this paper, we describe the implementation of a novel sensor network Android application called NodeDroid, which communicates with the sensor network base station to display processed sensor data and alerts. We chose to implement the multi-level security in the network because not all the nodes need the same level of security all the time. For example if a group of sensors just collecting the temperature and humidity they might not need the same level of security as the nodes which are collecting information like if there is any poison gas.

The paper is outlined as follows, section II provides the related work, section III describes the organization of the network, security of the network is explained in section IV, section V provides the design and implementation of the NodeDroid application. Section 6 describes the evaluation.

II. RELATED WORK

Thomson et al. [2] have proposed an accident detection system that can help emergency responders to reach to the accident scene faster through reduced response time. This is achieved by sensors and GPS embedded in any widely used smart phone. The detection process is done by the data from accelerometers and GPS. A mobile application named WreckWatch is used along with the proposed model. The proposed approach has some room for improvement. For example, it should be used in the application where the mobile phone does not have built-in sensor nodes. In our model, we do not assume that the mobile devices come with sensor nodes, rather we collect data from a sensor network and the mobile device communicates with the base station to display results.

Kapadia et al. [1] provided a model to process sensor network collected data in a cloud or grid and display the result through a GUI or internet browser. This approach is very useful for the applications where processing of data requires large amount of storage and significant CPU power not available in sensor networks. One of their goals is to filter out unnecessary and duplicate data and process the data considering security threats can happen in the network as well as in the cloud.

Miluzzo et al. [3] explained an approach that can be used to update the status of a person in a social networking site through the sensors in a smart phone. The sensors are going to sense the status through GPS, cameras, and other sensing devices. The authors claim this approach can work in Nokia N95 or iPhone. The paper provides the related issues with the hardware and software to achieve the goals. The major drawback of this paper is differentiating the events of a person by a sensor with the given data.

Boers et al. [4] discuss a platform for implementing sensor networks using the virtual component VUE2. Their goal is to optimize the hardware components to write software.

III. NETWORK ORGANIZATION

In this section, we provide the basic organization of the network. The organization is done using a Hasse diagram approach. To achieve that, the nodes in each group communicate with each other to find the node with highest energy, which is chosen as cluster head. The hierarchical organization of the network is done in top-down approach, starting from the cluster head (see Figure 1).

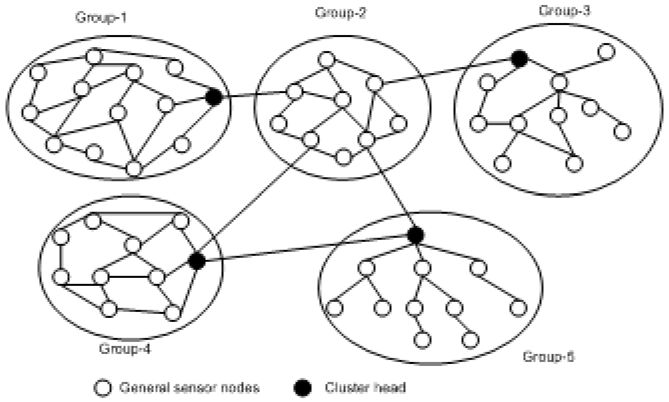


Fig. 1. Sensor network model.

We represent the sensor network by a graph G = (V, E), where V represents a set of sensors and E represents a set of communication links. A link between nodes (v, u) indicates that both nodes v and u are within the communication range. We assume that all the nodes are homogeneous. The corresponding graph is an undirected graph in which connections to nodes are determined by their roles and distances. The roles are assigned by the cluster head. The nodes with higher level of access try to reach the lower level nodes within one hop. If the higher-level nodes cannot reach the lower-level nodes in one hop, the higher-level nodes use intermediate nodes. The Hasse diagram algorithm uses marking of each node for organizing the network. For marking each node, the cluster head of each group runs an algorithm in the graph G = (V, E) starting from the cluster head. The notation m(v) is used as a marker for vertex $v \in V$ which is either T (marked) or F (unmarked). Initially, all the nodes in each group are unmarked and each vertex v has its neighbor set as N(v) = {u | (v, u) ∈ E}.

As an example, N(u) = {v, y}, N(v) = {u,w, y}, N(w) = {v,y}, N(y) = {u, v, w}. After the marking process, vertex u has N(v) and N(y); v has N(u), N(w) and N(y); w has N(v) and N(y); y has N(u) and N(v) and N(w).

The detail of this implementation can be found in our other research work in a protocol called RBASH [6].

IV. SECURITY THROUGH MULTI-LEVEL ACCESS CONTROL

In this section, we discuss secure information sharing using the hierarchical path. Our focus is on maintaining hierarchy rather than maintaining a shortest path. We seek to identify major approaches to achieve these goals like usage, revocation, re-dissemination, and distribution policy.

Usage policy: If a user is authorized to access data of a particular level in the network, then the system performs the requested data-sharing operations. Essentially there is no pre-defined usage control. The concept of limiting usage was first emphasized in recent years by RBAC where limits on how often or how long access is permitted are often viewed as a base of multilevel access. The usage policy in RBASH is based on the distribution of the key. If the higher level nodes have the keys used to derive the keys for the lower level nodes, then the higher level nodes have access to the lower level nodes.

It is necessary that the revocation must be addressed at the policy model. The revocation is handled by the implementation layer. Some of the questions addressed in this section are: Can authorized access be revoked? What is the delay in revocation? More generally, can authorized access be changed?

Figure 2 shows the authorization model. In this model, the authorization starts with the level selection of the network. At the time of choosing levels, the nodes can have their internal keys for the secure communication. A node can have different paths to reach to other nodes; it can choose any particular path based on the preference. The preference can depend on the number of hops. The role selection of the nodes is determined by three parameters: 1) a node's service history, 2) its current authorized tasks, and 3) expectation of other nodes. Each network, group, or region can have usage, revocation, and continuation policy determined by the higher level nodes. The output of this model includes: access control, key selection, and relationships creation.

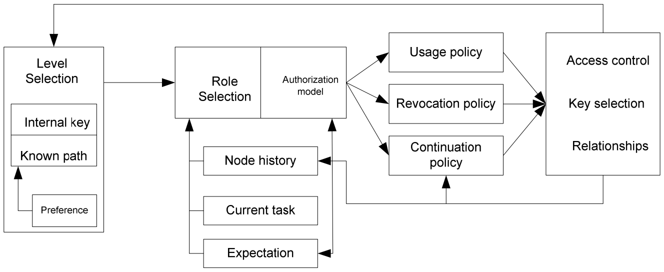


Fig. 2. Authorization model.

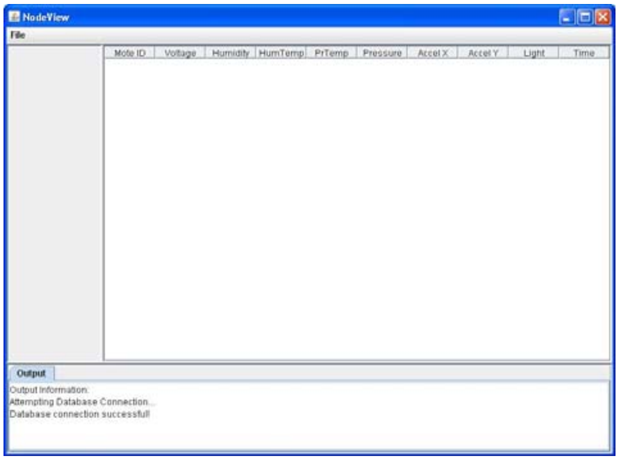
A. Level Selection

The level selection in the network or group is based on the internal key, known path, and preferences. The cluster head is at level 0. It has the key K_{1234}, which can be used to decrypt any messages of its group members. Node B can encrypt a message M using the internal key K_{123}. If $B[M]_{K_{123}} \to A$ message is sent from node B to A, then only A can decrypt it because it is in a higher level than B and has a direct communication link. The known paths are based on the key and the direct link.

The detail of this implementation can be found in our other research work in a protocol called RBASH [6].

V. NodeView

We chose to use the Java programming language because it is compatible with multiple operating systems, and would provide program portability [5]. Additionally, due to the many libraries available in Java, we can abstract a lot of the implementation details. The initial version of the GUI (see Figure 3) developed at our lab and named NodeView, provided the core capabilities necessary to the application. It established a serial connection with the sensor receiver unit, read the incoming packets, and converted the packet data. The sensor readings were displayed in a table and written to a SQL database for historical storage. The program contained a very simple interface consisting of the table, an output panel that displayed the status of program actions, and a menu bar containing the program commands.



Mote ID	Voltage	Humidity	HumTemp	PrTemp	Pressure	Accel X	Accel Y	Light	Time

```
Output Information:
Attempting Database Connection...
Database connection successfull
```

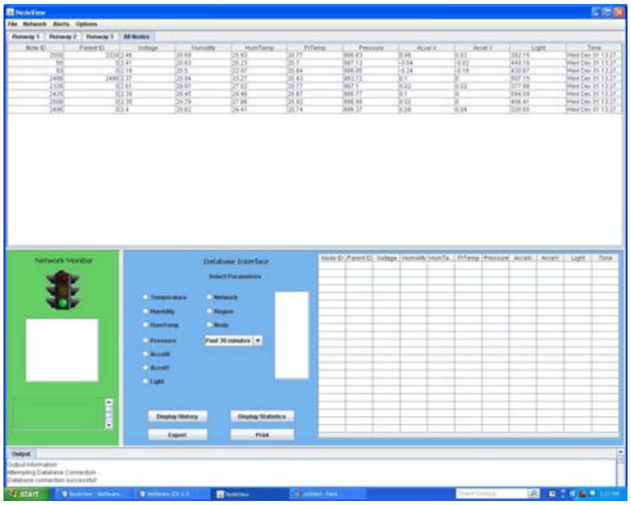
Fig. 3. Original NodeView GUI

The communications packets used by the sensors are a binary representation of hexadecimal values. Each packet is a string of bytes, with each byte containing one hexadecimal number. Pairs of bytes represent a reading for one of the variables being measured by the sensors. The two consecutive hex numbers are converted to decimal format and combined to generate one floating point value. This value must then be inserted into a conversion formula (specific to each variable) to obtain a decimal value in the appropriate measurement units.

A network structure of the nodes is necessary to organize and track incoming sensor data. We designed a structure consisting of three tiers: 1) network, 2) region, and 3) node. Each successive level is contained within the previous tier: The network is an array of regions, and a region is an array of nodes. Nodes extend the JLabel Java Swing component, which enables storing graphics data with sensor reading data. This data structure facilitated coding drag-and-drop graphics in the topology screen. The network structure is crucial, and is accessed by nearly every module of the program. In the event the user chooses not to manually configure a network, the program automatically configures one based on incoming sensor data.

The original version of the GUI stored sensor data in a SQL database, but provided no means to access it. To remedy this, we designed a database interface that was placed on a panel at the bottom of the main GUI screen. It allows the user to query the database using sensor variables as parameters. The user selects the scope of the query (network, region, or node), and can choose a pre-set time interval or enter a custom time range. On the right side of the panel is an area that displays a history table or statistics graph, based on the user's preference. Since users may require portability of retrieved data, options were provided to print the table/chart or export it to MS Excel.

One potential application of a sensor network is in an airport environment. Aviation safety could be increased by predicting the likelihood of ice-covered runways. In order to use sensor data in making such predictions, the GUI needed an alert system. The alert system we designed (see Figure 4) allows customizable alerts to be created by the user. The user can select up to two parameters for the alert, enter target values, and select the appropriate operators (i.e. equal to, greater than, less than). Notification of an alert being triggered can be displayed as a pop-up window or sent via email. An alert monitor panel on the main GUI also shows the status of active alerts. Every component previously added to the GUI was text-based. In order to provide a graphical experience for the user, we created the topology module. It opens in a separate window when selected from the program menu. It provides a graphic representation of the sensor network. Icons are displayed for each node. When the user rolls over an icon with the mouse, current sensor data and alert status for that node is displayed. The screen is customizable; allowing the user to drag and drop the icons, upload a background picture, and select a complementary color for the transmission path lines.



Output information
Attempting Database Connection...
Database connection successful!

Fig. 4. Fully developed NodeView GUI.

Sensor networks can conceivably cover large geographical areas, therefore NodeView needs the capability of sending and receiving data via the internet. Upon installation, NodeView can be configured to be a central server or remote client. Multiple NodeView installations can be linked into one network. This feature gives the user the capability to monitor a widely dispersed implementation of sensors.

In the process of adding the enhancements above, we quickly realized that multi-threading was a necessity for the program to function properly. The program must perform many operations simultaneously. Instances of the SerialComm and AlertMonitor classes run in always on/dedicated threads. When activated, the topology module creates a thread that periodically updates the display. On demand threads are created for database queries and the socket connections to remote NodeView installations and smart phones (see Figure 5).

```
class Node extends JLabel{
   …
   …
   public Node(){
             active = false;
               deci = new DecimalFormat("0.##");
             alerts = new ArrayList();
                         addMouseListener(new
           java.awt.event.MouseAdapter() {
                                    @Override
                          public void
mouseReleased(java.awt.event.MouseEvent evt) {
                             testNodeMouseReleased(evt);
                          }
               });

               this.setIconTextGap(-3);
               setVerticalTextPosition(BOTTOM);
             this.setHorizontalTextPosition(CENTER);
      }
       private void
testNodeMouseReleased(java.awt.event.MouseEvent evt) {
               windowRect = new Rectangle(xOffset,
yOffset+30, 1024, 728);
                      location = evt.getLocationOnScreen();
                      rect = new Rectangle(location.x - 18,
           location.y - 26, width, height);
               if (windowRect.contains(rect)){
                     setBounds(location.x - xOffset - 15,
location.y - yOffset - 52, width, height);
                      xCenter = location.x - xOffset;
                        yCenter = location.y - yOffset - 30;
               }
      }
…………
```

Fig. 5. Source code for drag and drop nodes.

A. Issues in Integration Process

Almost any desktop software can have some or all of its features incorporated into a smart phone application. Due to the recent and continuing growth in the smart phone market, creating a phone version of a software package is a smart move for developers [8]. It adds to the value of the program and creates alternatives for the consumer. However, coding a smart phone app requires a paradigm shift for the developer. Creating a robust, reliable phone application with a user-friendly interface presents new challenges.

Processor and Memory Limitations

Mobile phones generally have processors with slower clock speeds than a typical desktop or laptop computer. The RAM available in a phone platform is also smaller, and much of it is reserved for the phone operating system. The processor limitations are best addressed by shifting as much processing as possible to the server side. Also, algorithms in the application code should be streamlined, and designed in ways that minimize usage of the phone's memory.

Screen Size

Although smart phone screens vary in size, the real estate available to a developer is far smaller than what is available in a desktop application. One cannot simply scale down the existing desktop interface and display it on a phone screen. The small screen size may necessitate radically re-designing the original interface. Additionally, in some cases, phone operating systems don't support the libraries used to create the original desktop application, so the user interface must be coded to incorporate and utilize libraries provided by the phone platform.

Network Bandwidth

In recent years, telecommunications companies have taken great strides to increase bandwidth of mobile networks. However, 3G networks are congested in some geographical areas, and are non-existent in others [7]. The bandwidth available for a phone application's internet connection will vary with the user's location. Developers must keep this in mind, and if possible, design an application that isn't dependent upon large data transfers. Communication packets should be as size efficient as possible.

Overview of Android Platform

The Android operating system is built upon a Linux core. The Linux kernel performs low-level tasks for applications such as memory and process management [4][5]. Android apps are written in Java. An instance of the Dalvik virtual machine (similar to the standard Java VM) is created for each application upon initiation.

Tools

The Android platform provides several libraries for developer use in creating applications. These libraries include utilities for generating graphics views, media controllers, and local SQL databases, as well as others. Although Android doesn't support the Java Swing library, it does support many similar constructs. However, in our experience, the Android view library is not nearly as powerful as Java Swing. This requires developers coming from the Java Swing environment to find some creative work-around solutions in certain cases.

Framework

Android applications are built in a layered framework, and contain many files. The logic contained in an application, and the manner in which the application is presented graphically are distinct entities. Java class files contain code that implements application processes, but information to generate graphics layouts isn't hard-coded in them. Resources such as layouts and strings are defined in associated XML files. The Android Manifest XML file serves as master control for the program as a whole. Although this approach requires more effort in initially coding a program, it is beneficial in the long term. Future modifications to the app such as supporting multiple screen sizes and operating systems are simplified. Additionally, it aids in debugging and program maintenance.

Activity Class

The base component of an Android app is an instance of the Activity class. Applications have no main class; the Activity class is the equivalent of the main class in other Java programs. Each activity is defined in a Java class file. Activities have several predefined methods for implementing various states of the program. These methods are overridden by the programmer to define behavior specific to a particular application. Complex Android apps consist of multiple activities that are created, run, and terminated by user-initiated actions. An Android activity is analogous in some ways to the JFrame component of the Java Swing environment.

VI. Development of NodeDroid

When designing a smart phone version of an existing desktop application, one must first do a detailed analysis of the original program. Which features can be translated to a much smaller interface? What features of the original application are absolutely crucial to the program, and what can be omitted? Is it possible to split interfaces across two (or more) screens, but continue to provide a user-friendly experience? With these issues in mind, we took a critical look at NodeView.

Displaying current sensor data was a key feature that must be included. Without this feature, the application could not exist. However, displaying the sensor table of NodeView on such a small screen presented challenges. The network structure of NodeView would have to be copied to the Android version. As stated previously, most aspects of NodeView were dependent upon it. Without it, such features as database queries and topology display wouldn't function. However, the network couldn't be copied directly due to the Java Swing dependencies contained in it and the need to limit phone memory usage.

An interface to the database was a key component of NodeView, and a capability that should be offered on the phone version. Due to the small screen size, however, it would need to be re-designed. Rather than the interface and query results being displayed on the same window, they would have to be split across multiple screens. Access to the alert system was another feature that would be convenient for the user. Being able to monitor alerts, and having the ability to create, edit, or delete alerts via phone would enhance sensor network capabilities. If a topology screen were included, it would require a radical redesign compared to the original application. Due to time constraints imposed by the project deadline, we decided to omit this feature from the original version of NodeDroid.

Last, but certainly not least, is communication between NodeDroid and NodeView. Obviously, none of the components of NodeDroid could function without data from the NodeView server. As stated previously, due to mobile network bandwidth issues, communications packets would need to be designed to be as small as possible.

Implementation of NodeDroid

The initial step in constructing NodeDroid was to create the basic application shell. As stated previously, the entry point for an Android application is an instance of the Activity class. The base activity (NodeDroid) we created contains the menu structure to access sub-activities, and displays the program start-up screen.

Since communication with NodeView is essential, our next focus was establishing a socket connection and coding the communication packets. The connection code could be abstracted since both the Android platform and NodeView support the java.net.socket class. NodeDroid simply needed the IP address of the NodeView server to establish a connection. We designed two communications packets, a ClientRequest object and a DataUpdate object. The ClientRequest object to be sent from NodeDroid contained an integer field that indicated the type of data being requested, and fields to contain information updates for NodeView. The DataUpdate object was of similar design. Upon receiving the client request, NodeView would process a switch statement on the type, perform the requested action, then send the requested data back to NodeDroid via the DataUpdate object.

In order to pass these objects back and forth, both applications needed to utilize Java object input and output streams. When we created these streams and began testing the connection we encountered a problem. We could successfully open both streams, but when we tried to transmit packets, we would get a Java class loader exception. After many days of research and debugging, we located the source of the problem. The packet source code, which should have been identical down to the classpath, was being inadvertently changed. The root of the problem was using different integrated development environments (IDE's) in coding the two programs. NodeView had been created using Netbeans, but in order to utilize the tools in the Android special development kit (SDK), we had to use Eclipse. We had coded the packets in Eclipse, and when copying them to the NodeView computer, Netbeans was slightly altering the class path. It was a very simple issue to correct, but locating the source of it caused a substantial delay in progress.

Due to the fact that multiple screens would be necessary to port NodeView features to Android, NodeDroid would have to be a complex application consisting of several distinct activities. The Android platform doesn't provide a global memory space for activities. This presented a problem, because all of the activities would need access to the network structure and socket connection to NodeView. At first, it appeared that the only option would be to pass these two objects back and forth among the activities, which isn't a preferable style of coding a program. Additionally, the methods used to pass data between activities are designed for primitive data types. If we chose this route, we would have to use a parcelable interface to recreate the object each time an activity was called.

Fortunately, we found an alternative construct (see Figure 6). We placed the network and socket connection code in a wrapper class called NodeDroidApp, then created an instance of this object in the base activity. Activities called subsequently declare a local instance of the NodeDroidApp object, and set it equal to a function call of "getApplication()". The returned application was cast to a NodeDroidApp object. The local object could then access the required functions and data.

```
………
public void onCreate(Bundle savedInstanceState){
  super.onCreate(savedInstanceState);
  app = (NodeDroidApp) getApplication();
  setContentView(R.layout.sensortable);
  GridView gridView = (GridView)
findViewById(R.id.GridView01);
  gridView.setAdapter(new DataAdapter(this));
}
…
public View getView(int position, View convertView,
ViewGroup parent) {
     TextView tv = null;
     if (tv == null) {
         tv = new TextView(context);
        tv.setLayoutParams(new
        GridView.LayoutParams(43, 15));

//tv.setScaleType(TextView.ScaleType.CENTER_CROP);
        tv.setPadding(3, 3, 1, 1);
        tv.setTextSize(8);
        tv.setGravity(android.view.Gravity.CENTER);
        tv.setTextColor(Color.BLACK);
        } else {
          tv = (TextView) convertView;
        }
        tv.setText(app.data[position]);
        return tv;
}
…………
```

Fig. 6. Part of code for implementation.

Graphics Layouts

The first layout needed was for the sensor table display. Unfortunately, Android doesn't have a preconfigured data table view like the one in Java Swing. It does have a table layout that can be used however. We combined a table layout with gridview and textview objects, and produced a satisfactory data table display (see Figure 7). However, there are some drawbacks to this approach, which will be discussed later in the System Evaluation section.

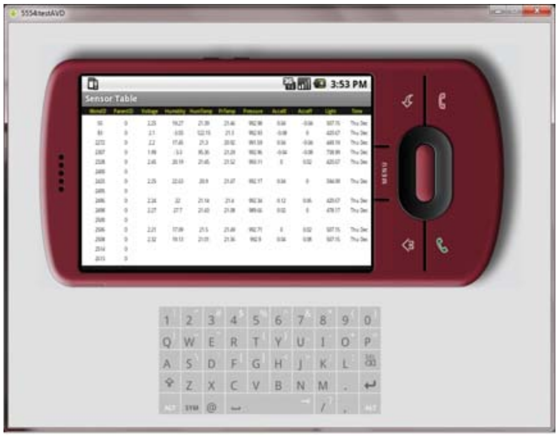
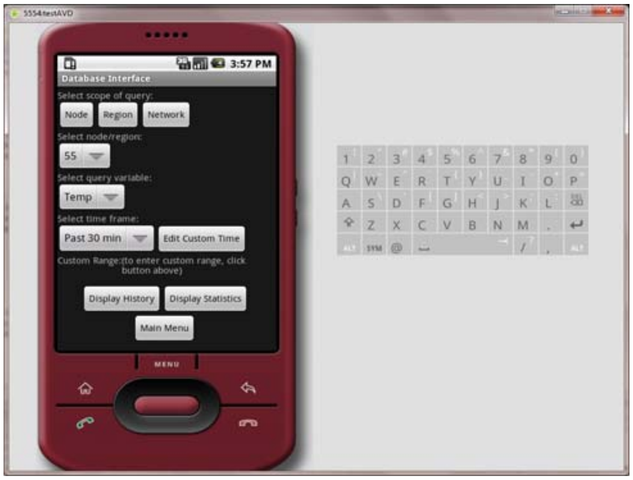


Fig. 7. NodeDroid sensor table.

Designing a database interface that provided the same features as the one in NodeView required a departure from the desktop design. There is simply no way to create a usable interface on a single phone screen that contains all of the elements of the original. The small screen size necessitated dividing the interface (see Figure 8) into multiple screens, utilizing pop-up list scrollers, and displaying the query result on a separate screen. The history data table contains the same fields as the sensor data table, so we were able to reuse that layout for displaying query results.



Database Interface

Select scope of query:

Select node/region: 55

Select query variable: Temp

Select time frame: Past 30 min

Custom Range:(to enter custom range, click button above)

Fig. 8. NodeDroid database interface.

VII. System Evaluation

NodeDroid was developed in phases. Modules of the program were coded, implemented, and tested before moving onto the next phase. The modules of the program that are functional at this time are reliable and user-friendly. When completed, the program will be a valuable addition to our existing NodeView software package. One potential drawback to the application, however, is the amount of data that can currently be pulled in database queries with long custom time ranges. In testing this feature at first, it seemed that the program had stopped responding. Further tests revealed that it simply was taking an inordinately long time for the phone emulator to process the data. The data set tested consisted of approximately 16,000 strings. In order to display the table, these strings had to be converted to individual text views, incorporated into a grid and then drawn on the screen. The amount of data being displayed would require the user to scroll the screen numerous times.

Displaying extremely large data sets on a phone screen is probably asking the Android platform to perform tasks that are unsuited to a mobile application. In order to ensure that NodeDroid is responsive, it may be advisable to limit the size of potential data sets. I think it's unlikely that a user would frequently desire to scroll through huge amounts of data on a phone screen.

VIII. Conclusion and Future Development

So far, it is assumed by most of the researcher that displaying the sensor data information in a base station computer is sufficient. With the sophistication of smart phones, it is convenient for the users to receive processed sensor data on a smart phone. In this paper, we have described the implementation of a sensor network Android application called NodeDroid, which communicates with the sensor network base station to display processed sensor data and alerts. We chose to implement the multi-level security in the network because not all the nodes need the same level of security all the time. For example, if a group of sensors just collect temperature and humidity data, they might not need the same level of security as nodes collecting information such as presence of poisonous gas.

Due to the variety of Android smart phones currently available, there are multiple versions of the operating system in use. Compatibility with all of them must be assured prior to releasing the application. Since Android is not the only smart phone platform, after completing and releasing NodeDroid, the application should be rewritten in Objective-C. This will allow it to be ported to the iPhone. Having versions for both Android and iPhone will make NodeDroid available to the largest possible market.

References

[1] A. Kapadia, S. Myers, X. F. Wang, and G. Fox, "Secure cloud computing with brokered trusted sensor networks," in *Proceedings of the 2010 International Symposium on Collaborative Technologies and Systems (CTS 2010),* Chicago, IL, USA, May 2010, pp. 581-592.

[2] C. Thompson, J. White, B.Dougherty, A. Albright, and D. C. Schmidt, "Using smartphones and wireless mobile networks to detect car accidents and provide situational awareness to emergency responders," in *Proceedings of the 3rd International ICST Conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications (Mobilware 2010)*, Chicago, IL, USA, June/July 2010, pp. 29-42.

[3] E. Miluzzo, N. D. Lane, K. Fodor, R. A. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell, "Sensing meets mobile social networks: the design, implementation and evaluation of the CenceMe application," In *Proceedings of 6th ACM Conference on Embedded Networked Sensor Systems (SenSys '08),* Raleigh, NC, USA, November 2008, pp. 337-350.

[4] N. M. Boers, P. Gburzyński, I. Nikolaidis, and W. Olesiński, "Developing wireless sensor network applications in a virtual environment," *Special Issue: SIS 2008, Telecommunications Systems*, vol. 45, no. 2-3, pp. 165-176, 2008.

[5] R. Rogers, J. Lombardo, Z. Mednieks, B. Meike, *Android application development: Programming with the Google SDK*, Sebastopol, CA: O'Reilly Media, Inc., 2009

[6] B. Panja, S. Madria, and B. Bhargava, "A role-based hierarchical sensor network architecture to provide multilevel security," *Computer Communications Journal on Algorithmic and Theoretical Aspects of Wireless Ad Hoc and Sensor Networks,* vol. 31, no. 4, pp. 793-806, March 2008.

[7] K. Han, J. Kim, K. Kim, and T. Shon, "Efficient sensor node authentication via 3GPP mobile communication networks," in Proceedings of the 17th ACM conference on Computer and Communications Security (CCS '10), Chicago, IL, USA, October 2010, pp. 687-689.

[8] P. Andreou, D. Zeinalipour-Yazti, P. K. Chrysanthis, and G. Samaras, "In-network data acquisition and replication in mobile sensor networks,"*Distributed and Parallel Databases,* vol. 29, no. 1-2, pp. 87-112, February 2011.

Security of Computer Use Practice: The Case of Ordinary Users Survey

Leon Reznik, Vincent J. Buccigrossi III, Justin Lewis, Asif Dipon, Stefanie Milstead, Nathan LaFontaine, Kenneth Beck, Holden Silva

Abstract—**While numerous surveys in regards to computer security are conducted by organizations in order to study the behavior of their employees and users, the most influential in determining the security of a general computer environment group of ordinary users is still lacking in attention. This paper presents the survey results of more than 3,000 computer users who participated in a survey conducted in 2010-2011 by Prof. Reznik's students at RIT. They researched the practice of home computer users including: the passwords used, patching regularity, firewalls, and anti-virus software application. The samples of results are analyzed and presented here. These results may be utilized for improving overall computer security as well as for the education of users.**

Index Terms—**Computer Security, Data Collection, Survey**

I. Introduction

THIS paper attempts to evaluate the security of home computer use practice based on the findings from the survey conducted by Prof. Reznik's students in the class "Security Measurement and Testing" at the Department of Computer Science, Rochester Institute of Technology, New York. Both graduate students in the MS program in Computer Science and undergraduate students in the BS program of Computer Science, as well as one student from the Information Security and Forensics B.S. degree program participated in this project in the winter quarter of 2010-2011. Students planned and conducted the survey, processed, as well as analyzed the results. A sample of the results and their analysis are presented here.

Neither the idea, nor the practice of surveying computer users in relation to various aspects of computer security is new. The well known CSI/FBI Computer Crime Survey has been conducted for fifteen years [1]. Academic publications are available [2]. Others attempt to target mainly the personnel of certain corporations and other organizations with the goal to improve security of a particular information system's infrastructure. Surveys aimed at ordinary users are much more rare. Those known examples were conducted by university teams. For example, one conducted by Columbia University studied ordinary citizens' attitude towards privacy while working with various websites [3]. It collected data from only 159 respondents regarding very specific questions. Another one, conducted by Pennsylvania State University [4] focused on consumer and end-user level information security and attracted 368 responses, but again was only devoted to electronic commerce infrastructure.

This survey concentrates on studying the computer practices of ordinary users who usually administer their own home computer environments. However, their interconnection with the rest of the computer world makes learning the characteristics of their computer practice extremely important. In this survey, data was collected regarding ordinary user security practices including: firewall usage, antivirus software, and patching regularity of operating systems. Targeting the survey at a wide demographic of ordinary users, as well as the large sample size (more than 3,000 responses from various groups collected by different means -- See Section II for more details on respondents' demographics) make the results quite unique. The bulk of the questions were devoted to the cornerstone of computer security – passwords. Section III presents the analysis of passwords in relation to user's age, gender, educational level and general computer expertise (subsections A-C). Also, the survey tries to find if there is any difference between critical and non-critical passwords, e.g. financial institutions' passwords versus a user's website forum password. Further analysis of a general computer practice security is also presented in section III, subsections D-E.

II. Respondents Pool Demographic Information

Three different methods of collecting information were employed: 1) posting the questions to the chart group on Reddit (special group of mainly computer educated users), 2) posting the questionnaire into a page in a social network such as Facebook, and 3) using personal connections by asking friends and family to respond to the survey. No financial or any other incentives were offered to respondents for completing the survey. Three student groups selected one method of data collection each. In terms of number of responses, the Reddit was much more successful than the other two groups. The Reddit survey had about 3,000

L. Reznik*, V. J. Buccigrossi III, J. Lewis, A. Dipon, S. Milstead, N. LaFontaine, K. Beck, and Holden Silvia are with the Department of Computer Sicnece, Rochester Institute of Technology, Rochester, NY 14623 USA (is with the National Institute of Standards and Technology, Boulder, CO 80305 USA (corresponding author e-mail*: lr@cs.rit.edu).

responses, while the other two surveys had between 100 to 300 respondents. From those surveys, the data were slightly biased to the student's age group and education level as a lot of the surveys were spread through personal social media networks. The respondent demographics in terms of gender and education are presented in Table I.

TABLE I
DEMOGRAPHICS BREAKDOWN

Gender	
Male	2753
Female	239
Education	
Graduated college (computer related field)	746
Graduated college (non-computer related field)	496
Some college (computer related field)	857
Some college (non-computer related field)	520
Graduated high school	188
Some high school	188

III. Results and Analysis

A. Password Complexity and Age

From a high-level perspective, the password practices of the individuals that were surveyed followed no specific trend line. The vast majority (just over 33%) of those surveyed claimed to use strong passwords composed of all three attributes measured (numbers, alternating case, and symbols -- See Fig.1). However, that was only 3% higher than the next largest group – passwords that use both numbers and alternating case (29%). The third largest group appears to prefer only numbers (24%) for its password security.

Age appears to play a factor in the password complexity of users. The results for this specific category also follow no trend line, although it appears that older individuals (35+ years) make use of less-secure passwords, with 44% of respondents saying they utilized only numbers in increasing their password complexity. There are a couple of explanations for this type of behavior. The first explanation for this is that the majority of people in this age group was not raised in a computer-centric era, and thus, was not exposed as early on to the necessity of strong passwords. Another explanation could be that the group favors passwords that are easier to memorize, to avoid writing passwords down. This may be better for their security practices, though it would still be highly beneficial for them to use even slightly more complicated passwords.

People between the ages of 26 and 35 seemed to be more security-conscious than those under the age of 25, with 40% of survey-takers in that age range using the highest-complexity passwords. Within that age group, individuals over 30 years of age tended to favor using only numbers (30% vs. 20%), though 20% of 31 to 35 year olds made use of special symbols and numbers in their passwords and 30% of 26 to 30 year olds used both numbers and letters of alternating case.

Only 35% of survey-takers under the age of 25 used highly complex passwords; this is still a significant amount compared to people over 35 years of age (with 18%). The vast majority of these surveyors, however, use only numbers and alternating case for their passwords (an approximate 45%).

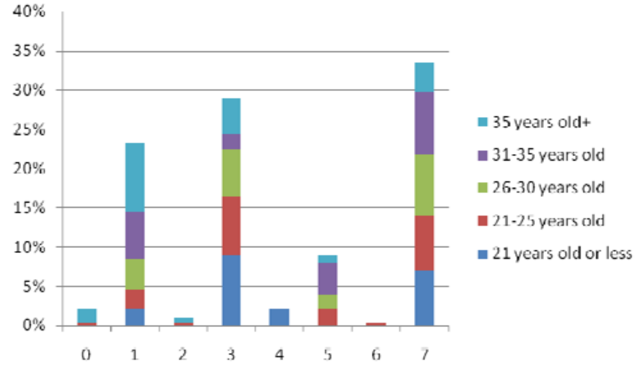


Fig. 1. How does overall password composition look? Key: 0 = Only lower-case letters; 1 = Numbers; 2 = Upper / Lower-case; 3 = Numbers + Alternating case; 4 = Special characters; 5 = Special + Numbers; 6 = Alternating case + Special; 7 = Numbers + Alternating + Special

Note that the above results did not take into consideration several other factors that we measured. These other factors included password length (the majority of users, of all ages, use passwords between 7 and 9 characters in length), whether or not users write down their passwords (vast majority of those under the age of 35 said they didn't, while 50% of those over 35 said they did), and whether personal information was included in a user's password (again, the vast majority of people under 35 years of age said they did not, while 44% of people age 35 and up said they did). Given this extra information, it can suggest the confirmation of the original statement that older individuals favor less-secure passwords and partake in less secure practices. Despite the earlier statement that they may choose less-secure passwords so that they can more easily memorize them, it appears as though these passwords are still written down. As for the rest of the age groups, it is difficult to conclude which age range used the best password security practices, but with 40% of people between the ages of 26 and 35 including numbers, alternating case, and symbols in their passwords, they might have the greatest number of secure users.

B. Critical Passwords vs. Non-Critical Passwords

This section describes the passwords that respondents use for critical applications, such as financial institutions (Table III), web access in comparison against non-critical password-protected sites and systems (Table II). The possible choices for this question included numbers, mixed case letters, special characters, personal information, and phrases. This characteristic is difficult to graph and match with others, but it is still interesting to note the most popular password trends. The top 5 most popular and least popular choices based on the analysis of the data are shown.

As one can see, critical password usage trends are analogous to the results for non-critical passwords. The common combinations are also similar as the numbers are comparable. The less common choices do not differ by much either, with personal information mixtures being the least popular overall. This also means that users’ critical passwords suffer from the same problems as users' non-critical passwords. Different results were expected, however, given the fact that banks tend to restrict password length and special characters. These results can be seen as a potential positive, though, due to the common trends of banks using personal information to identify their clients, perhaps explaining this trend. The cause is not the lack of user awareness, but the mandate for personal information use as the security credentials to prove user authenticity.

TABLE II
NON-CRITICAL PASSWORD GENERAL
COMPOSITION AND COMMON COMBINATIONS

Mixed Case	2151
Personal Information	765
Numbers	2747
Phrases	1047
Special Characters	1568
Mixed Case, Numbers, Special Characters	986
Mixed Case, Numbers	687
Numbers	370
Mixed Case, Numbers, Special Characters, Phrases	197
Numbers, Special Characters	142

TABLE III
CRITICAL PASSWORD GENERAL COMPOSITION AND COMMON COMBINATIONS

Mixed Case	2675
Personal Information	193
Numbers	2786
Phrases	425
Special Characters	1524
Mixed Case, Numbers, Special Characters	1115
Mixed Case, Numbers	713
Numbers	367
Mixed Case, Numbers, Special Characters, Phrases	174
Numbers, Special Characters	92

Overall, these results are not pleasing. Only about 1/3 of respondents have satisfactory password habits (not counting length). Considering our survey did not check any boxes automatically, the fact that numbers alone are used much more frequently than even letters alone is troubling. Upon viewing this question, it is clear that a choice is missing, being dictionary words or single-case letters. It is possible that respondents who used one case and numbers in their passwords did not click “mixed case” because it technically was not true, which would inflate the 'numbers alone' category.

Unfortunately, a large portion of respondents seemed to choose unfavorable passwords. Given the previous results on a password length, an assumption can be made that the average numbers only password is at a length that is trivial for an attacker to defeat. It is also important to note that while the least popular choices all include personal information, which can be seen as a good thing, the individual categories 'Personal Information' and 'Phrases' alone are more popular than a mixture of them. In practice, this means that if users are predictable, attackers have an even easier time guessing their favorite phrases or pets.

C. *Password Choice Trends*

In this section, common patterns of password security are studied. Figure 2 presents the password length change versus the respondent’s age group and Figure 3 versus an education level. Further analysis demonstrates the presence or the absence of good password habits, such as changing passwords regularly. Figure 4 shows the dependence of a non-critical password changing interval on age and Figure 5 on education.

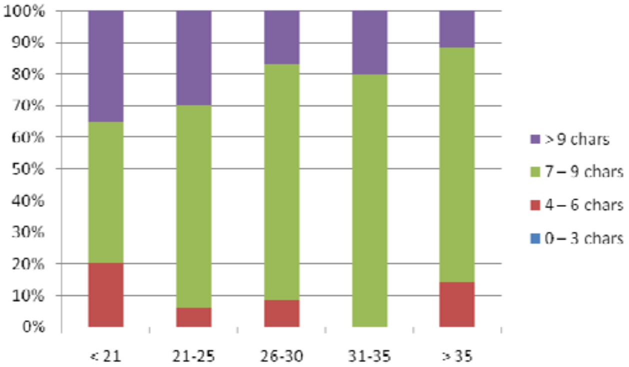


Fig. 2. Password length by age.

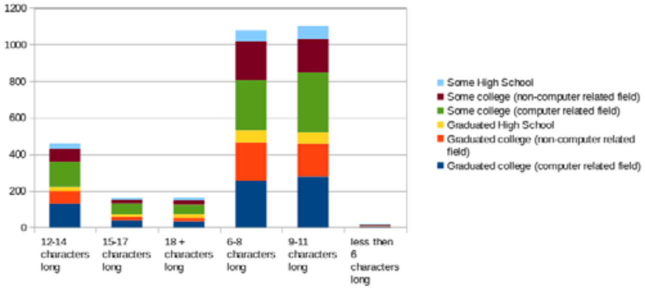


Fig. 3. Password length by education.

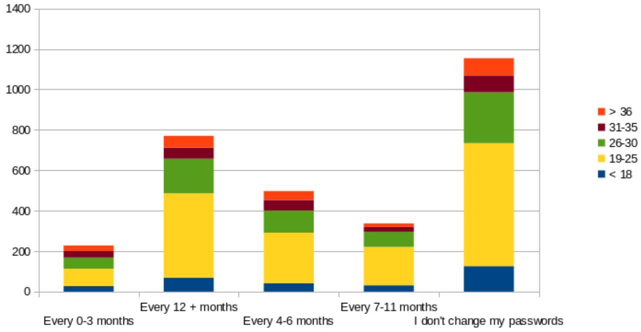


Fig. 4. Password change frequency by age.

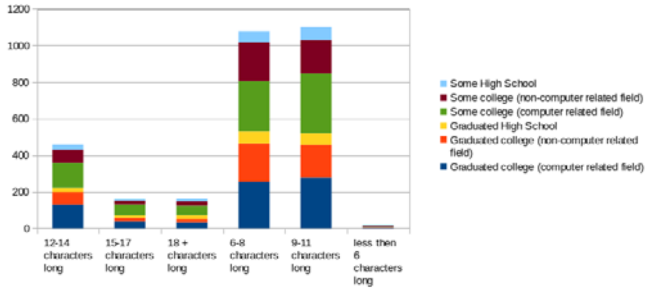


Fig. 5. Password change frequency by education.

Figures 6 and 7 demonstrate financial password reuse by age and education. For the next comparison, an aggregate score was derived comprised of the respondents' reported password security practices when compared to their self-described computer expertise. When asking them to describe their computer expertise, a described rating of '1' indicataes an absolute novice when it comes to computer use while a rating of a '10' would be a computer virtuoso. To come up with a rating for their password security practices, data was combined from each of the survey questions asked about password security. Responses that fell in line with secure password standards resulted in a higher password security rating for a respondent.

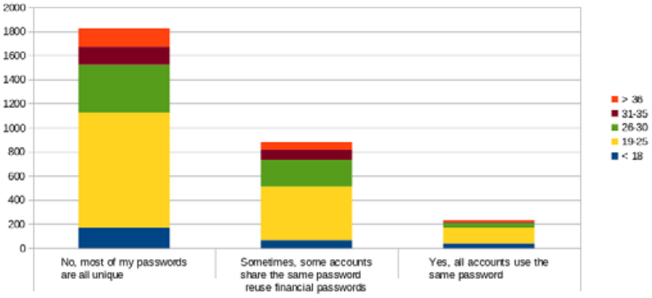


Fig. 6. Financial password reuse habits by age

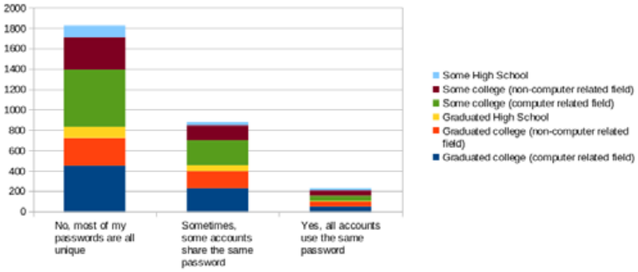


Fig. 7. Financial password reuse habits by education.

In Figure 8, the results that have been rendered with the analysis of the data found that the most prevalent answer by all age groups was that they do not change their password. This may come as a surprise as most sites require a regular password change. This may be the reason for the rise in responses in the 4-6 and 12 month categories.

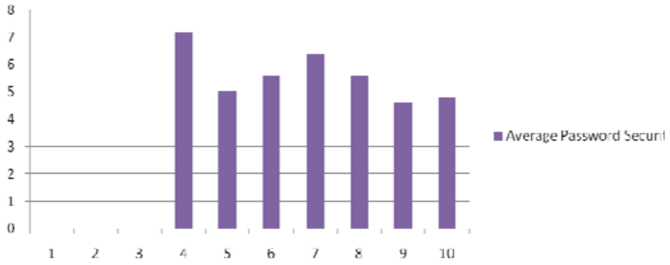


Fig. 8. Password security vs. computer expertise.

When comparing password security rating to reported computer expertise of the respondents (Figure 8), many may be surprised to find that the results contradicted the assumption nearly entirely. The assumption had been that there would be a direct relationship between computer expertise and password security, meaning that as computer expertise increased for a respondent, password security rating would also increase. It was suspected that users who used safe passwords and were aware of the practices that would encompass safe password security were more likely than not to be expert computer users in other areas. For example, doing research to learn about safe password security practices requires having some basic knowledge about computer security to begin with.

When the data are actually analyzed, the results for this comparison, however, may be surprising as findings show that the opposite actually seemed to be the case. Respondents with high computer expertise, instead of reporting very secure password practices, reported the lowest level of password security. Similarly, respondents who claimed a very low level of computer expertise actually reported practicing the safest of password practices. This ran completely counter to the assumed outcome, since the guess had been made that there would be a direct relationship and instead there seems to have been an inverse relationship among our respondents.

This might be explained though by a very simple, but very serious occurrence. Since respondents who reported themselves as being computer savvy also reported using weaker passwords and poorer password practices, this leads one to question if computer savvy users believe they are immune to attackers because they feel they are more educated.

Some future questions on this topic would be to ask for more details about whether respondents were required to use safe passwords as part of a business or website requirement. This would allow analysis of whether the cause of the less advanced users being more likely to use secure passwords may have been due to their occupations and corporate security policies. This might be derived by looking at the relationship between the computer expertise and the operating system used (Figure 9) and the browser chosen (Figure 10).

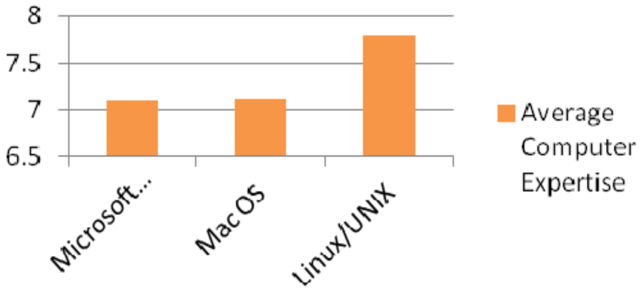


Fig. 9. Computer expertise relationship vs. operating system choice.

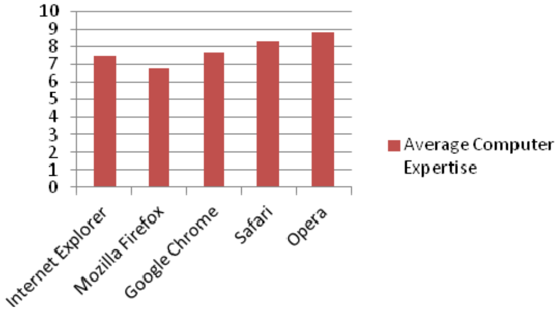


Fig. 10. Browser choice relationship with the operating system device.

D. *Operating Environment vs. Computer expertise*

When comparing operating system choice to reported computer expertise of our respondents, the results of the data lined up to fit the expected results. The speculation had been that respondents who used Microsoft Windows or Mac operating systems would also be users who felt less confident about their computer usage and would thus report lower levels of computer expertise. On the other hand, it was suspected that users of the Linux and/or UNIX operating system variants would be much more likely to be expert computer users as these operating systems generally require a much higher level of sophistication from their users. Additionally, users often have to make a choice to switch from the stock operating system, which is usually Microsoft Windows or Mac OS, to a Linux operating system. Being able to make the choice to switch operating systems and having the confidence to use the Linux operating system would require a level of expertise on the part of the respondent that we reasoned would lead them to report a higher level of computer expertise.

When analyzing the results for this comparison, it was pleasing to find that the operating system chosen by the respondents with the highest average levels of computer expertise was in fact Linux/UNIX. This fit the previous statement exactly, since the guess was made that Microsoft Windows and Mac OS respondents would be less expert. When comparing the expertise of respondents using Microsoft Windows and Mac OS, there was very little difference. This was somewhat surprising, as it was suspected that maybe the Mac OS users would be more likely to consider themselves expert computer users, but still fell in line with our hypothesis that the users of Linux/UNIX operating system variants would be more experienced.

Some future questions to explore surrounding this topic would be to ask for more details about which variants of Linux the respondents selecting Linux used. This would allow further analysis of whether some of the more specialized Linux operating systems were drawing more expert users. Additionally, one may wonder which OS would enable a more accurate computer expertise rating without relying on respondents to assess their own ratings.

In comparing browser choice to reported computer expertise of the respondents, we once again came to somewhat surprising conclusions. The original thought had been that respondents who used Internet Explorer or Safari would also be users who felt less confident about their computer usage skills and would thus report lower levels of computer expertise. Conversely, it was suspected that users of Firefox, Chrome, and Opera browsers would be more likely to assess themselves as expert computer users as these browsers offer much more customizability and extensibility when compared to Internet Explorer or Safari. Additionally, users have to make an active choice to switch from the stock browser which is Internet Explorer in Windows or Safari on Macintosh. It was reasoned that the ability to make the choice to switch to a different browser as such would be made by a more computer savvy respondent who would not have any issues with the installation or transition.

It was surprising to find that the browser chosen by the respondents with the lowest average reported levels of computer expertise was actually Mozilla Firefox. This ran counter to our original hypothesis that proposed that Internet Explorer users would be among the least experienced. After Firefox, Internet Explorer and Chrome had mid-range computer expertise ratings which at least somewhat confirmed the hypothesis that Internet Explorer would be among the lower end of the reported expertise range. However, the Safari web browser had the second highest average computer expertise rating which was very surprising. Unsurprisingly for most, Opera was the browser with the highest computer expertise rating. Opera is typically used by a niche set of users who were suspected to be savvy computer users since it is not that commonly heard of. It was reasoned that the differences between the original hypothesis and the respondents' reported browser choice may be that of an underestimation of the bias that the respondents may be giving to their own expertise. Despite what an objective observer may have concluded, users may have been likely to exaggerate their own computer expertise despite actually being novice users. Given the fairly vague nature of the expertise rating, it is unsurprising that respondents were unable to accurately judge their own computer expertise. Another additional issue may be that the survey failed to include an 'Other' category, which may have captured respondents who were unaware of which web browser they were using.

Additionally, the exploration of questions which would enable one to ascertain a more accurate computer expertise rating without relying on the respondents to assess their own ratings. Relying on respondents to assess their own computer expertise rating may have resulted in users coming up with a variable rating which in effect may not have been a good measure of their actual computer expertise.

E. Security vs. Age and Computer Expertise

For this conclusion, the average system security rating of the respondents was analyzed and compared to the respondents' reported age. To come up with an average system security rating, each user was awarded one point for each piece of software they used from the following list: firewall, antivirus, and anti-spyware software. The lowest possible security rating was a 0 for a user who chose not to use any security software and a 3 for a user who chose to use all three pieces of security software asked about in the survey.

The prediction had been that the younger the users were, the more secure their system would be, based on the assumption that younger users would tend to be computer literate. Such a user would therefore be more aware of the security concerns addressed by using firewalls and antivirus/anti-spyware software. Respondents under age 21 all reported that they used all three possible pieces of security software; however, users between the ages of 21 and 25 scored the lowest on their average security rating, with most reporting the use of around 2 out of the possible 3 pieces of security software. Contrary to the expected findings, the oldest age group, ages 35 to 50, scored nearly as high as the youngest age group when rated for average system security. Justification for this finding may be that given their relative inexperience in general as a group, older respondents were possibly more frightened by the dangers posed by their computers and thus more likely to seek out protection and security.

Some future questions to explore to be able to further analyze this data would be about the reasons respondents use computer security software. Being able to know the reasoning for their use would allow further insights into the reasons behind the results.

When comparing users' ages (Figure 11) and their computer expertise to their average system security rating (Figure 12) and network security rating (Figure 13), it was expected that users with higher computer expertise would have a higher average system security ratings and lower computer expertise would result in lower average system security ratings. However, it was surprising to see that users who reported a computer expertise in the range of 6 to 7 had the highest average system security ratings. It was expected that users who ranked themselves as "computer virtuoso" would have an average system security rating of approximately 3, but astonishingly it was found that they had an average system security rating of less than 2. The lowest computer expertise reported was 4 associated with the lowest average system security rating of approximately 0.6.

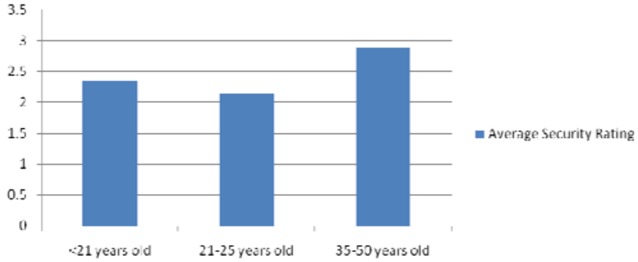


Fig. 11. Age vs. average system security rating.

From these results, it was concluded that users who ranked themselves in the mid range of computer expertise took greater precautions when deciding their system security software and used a greater variety of security software as opposed to users who ranked themselves at a lower range who had low average scores on the system security ratings. The hypothesis concerning users with higher computer expertise was not validated by our results. It is suspect that this may be caused by vagueness of the concept of computer expertise and users were overestimating their computer expertise. This is most likely what led to their unexpected lower average scores on the system security rating with an averagse of less than 2.

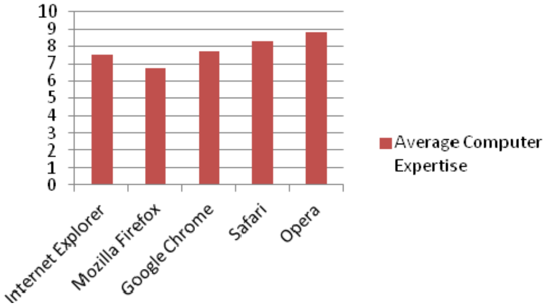


Fig. 12. Browser choice relationship with the operating system device.

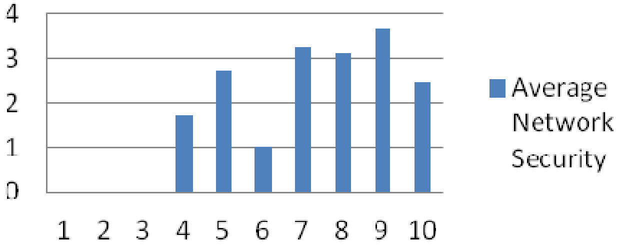


Fig. 13. Computer expertise vs. average network security.

To come up with an average network security rating, it was decided to give each user one point if they used a network device such as a router or hub in the network and one point if they enforced a security policy on their network. It was also decided to give them one point if they had a wireless network that used WPA, two points if they had a wireless network that used WPA2, and zero points if they had an open wireless network or no wireless network device. The sum of these values was then used to determine the user's average network security rating. A sum of 0 indicates that the user does not use any network device. A sum of 1 indicates that the user utilizes a network device with no security on it. A sum of 2 indicates that the user has a network device that has an open wireless network. A sum of 3 indicates that the user has a wireless network device using WPA. Lastly a sum of 4 indicates that the user has a wireless network device using WPA2.

When comparing users' computer expertise with their network security rankings, we initially thought that users with high computer expertise would also have high network security rankings. It was also expected that users with mid range computer expertise would have relatively high network security rankings because it is a well discussed topic in mass media due to its connections with legal matters (i.e. piracy and open wireless networks) and that most businesses that use networked computers enforce a network security policy.

The results helped to confirm that users in the mid range for computer expertise (5-8) mostly had high average network security ratings; all of them were above 2.7 out of the maximum of 4. The only exception to this was that users with computer expertise of 6 had the lowest average network security ratings of approximately 0.9. Users with high computer expertise did not perform as highly as expected, while users with computer expertise of 10 had a relatively low 2.4 network security rating. However, users with computer expertise of 9 had the highest network security ranking of 3.6.

These results lead us to conclude that the majority of users who fall in the mid range of computer expertise are using a network device with some form of security policy, which they may or may not be able to identify. Since users had the option to state that they did not know what security policies might be in place, this would account for their average network security ratings all being above 2.7. Lack of knowledge about their network and its security may also be the reason for users who had computer expertise of 6 to have such a low average network security rating. It was seen that average network security rating of users with computer expertise of 10 was relatively low and that leads to a conclusion that many users overestimated their computer expertise. Again, the vagueness of the concept of computer expertise may be responsible for this result.

F. System Security vs. Network Security

In this section, the average score of the respondent's system security (Figure 14) or network security (Figure 15) rating in comparison to their operating system was analyzed. The survey allowed respondents to select: Mac OS, Microsoft, or Unix/Linux as their possible operating systems because these are the most commonly used operating systems.

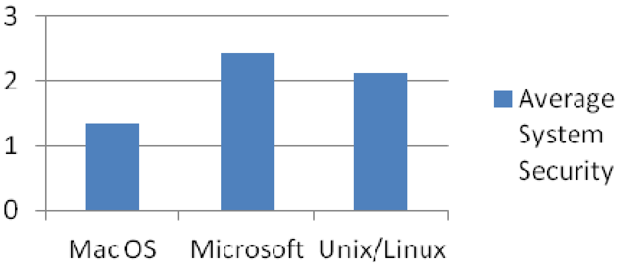


Fig. 14. System dependence on the operating system.

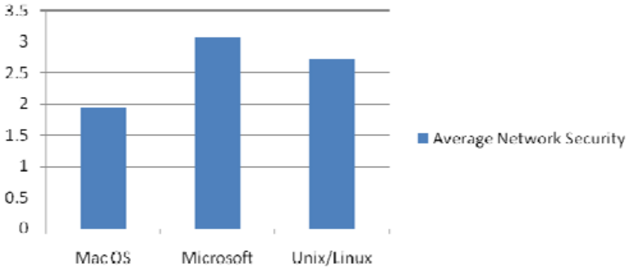


Fig. 15. Network security dependence on the operating system.

To come up with an average system security rating, each user was given one point for each piece of software they used from the following list: firewall, antivirus, and anti-spyware software. The lowest possible security rating was thus a 0 meaning that the responded used none of the listed security software. A rating of 3 meant that the respondent used all three pieces of software.

When comparing a user's operating system with their average computer security rating, it was assumed that respondents running a Unix/Linux operating system would have the highest ratings. This is because it was thought that due to the extra knowledge that it takes to run Linux/Unix and also due to the fact that their operating system is much more configurable at the installation. It was also expected that Mac OS users would have the lowest, as there are limited options in terms of security software for them and many Mac OS users feel that they have less security concerns in terms of viruses and malware. It was thought the ratings for Microsoft users would fall in the middle ranges due to the fact that Microsoft has treated system security as a major issue in their latest operating systems due to being the target of numerous exploits, viruses, and other malicious software over the past years.

The results demonstrated that Mac OS users scored the lowest in terms of average system security ratings. Surprisingly though, Microsoft users had a much higher rating than Unix/Linux users with a difference of approximately 0.3. It was theorized that the reason for this occurrence is because Linux/Unix users may feel safer using their operating system and feel that using a layered approach to security is not necessary. What these results have been able to show is that most Microsoft users are being proactive in the approach to security and using at least two different applications for security. This is probably due to the free Windows Firewall that is included with all Microsoft operating systems and the many varieties of free anti-virus software available for Microsoft users. The results lead one to believe that anti-spyware software is not used by most users of all operating systems listed because it is not deemed as necessary for security. There is also the possibility that the users are not aware of all the security software that is installed on their computer system.

From the results, one may conclude that Microsoft users are more aware of the security policies in place on their network and take more network security precautions. This may be caused by the increased awareness of the need for network security that is constantly highlighted in the media because of privacy concerns (confidential information being illegally accessed over networks). It was a surprise that Unix/Linux users did not score better. Their relatively low scores may be caused by many of their users being unaware of their network security policies or deciding to use network security measures that were not taken into consideration when building the survey. It also leads one to conclude that being knowledgeable about computers does not imply knowledge of networks. Mac OS users performed as expected, helping to validate the assumption that Mac OS users either are unaware of their network security policies or they do not feel at risk from network-based threats to their operating system.

IV. Conclusion

In the corporate world, computer security policies are put in place to help ensure that all users are following security best practices, but in the world of personal computing there is no one administrating the user's home computer network. For this reason, computer security of the general public is a growing concern in the increasingly connected world. Everything from passwords to firewalls needs to be used not only in the corporate workspace, but also in personal or private home networks. The survey presented here helps to provide insight into the common practices of the average computer user and examine what areas of computer security personal computer users should focus on. Using this information, one may be able to identify problem areas, and help to provide corrective action.

With the information obtained here, one can see the flaws in the average personal computer security practices. These flaws are what can be used to help educate the general public on the issues with their security. For example, the information gathered showed that most users use the same password for multiple logins. Even though those surveyed can claim that they only use the same password when the site is non-critical, it still constitutes a security risk. Other examples include the high number of users that use only numbers in their passwords, or the relatively small number of older users that used mixed-case and special characters in their passwords. This information can also be valuable to attackers; by reading this, an attacker now also knows the weaknesses in the average persons' personal security practices and can use this information against them. It is important to deploy surveys like this and use the information gained to help educate people on their security needs before an attacker exploits their weaknesses. One positive thing to note from the survey is the relatively high system security score of all users surveyed. On a scale of 0-3 where 3 was the highest possible score for system security based on the information gathered, every age group examined averaged higher than a 2. This means that most users are employing some combination of firewalls, anti-virus, or anti-spyware, which should help thwart would-be attackers.

The data has provided the information above on the issues many people have with their general password and computer security, and it is anything but perfect. A few things must be noted when examining the data. First, a large portion of the information collected came from male users of a popular social media site. Because of this, one must realize that any data analyzed in an over-reaching fashion would contain a great male computer user bias. That being said, when analyzing the data with respect to other demographics, such as age, or more importantly gender, the information gathered becomes much more compelling. Another thing to note is that the information gathered is from three different surveys collected through different means. While unlikely, this could mean duplicates exist across the three surveys. Furthermore, the three surveys contained varying wording on similar questions which then needed interpretation, and as such, the results presented here is one interpretation. Other than these minor flaws, the data collected from the three surveys meshed well together and the information presented here can be used for both educational and analytical purposes.

References

[1] Computer Security Institute [Online]. Available: http://gocsi.com/survey.

[2] M. Zviran, and W. J. Haga, “Password security: an empirical study,” *J. Management Information Systems,* vol. 15, no. 4, March 1999, pp. 161-185

[3] S. J. Stolfo, E. Johnson, and T. Pavlicic, “Citizen's attitudes about privacy while accessing government and private websites: results of an online study,” in *Proceedings of the 2003 Annual National Conference on Digital Government Research*, Digital Government Society of North America, pp. 1-6

[4] J. B. Gross, and M. B. Rosson, “End user concern about security and privacy threats,” in *Proceedings of the 3rd Symposium on Usable Privacy and Security* (SOUPS '07). ACM, New York, NY, USA, 2007, pp.167-168

Keynote: Working to Achieve Cybersecurity in the Energy Sector: A Public-Private Partnership Approach

Diane T. Hooie
Deputy Assistant Secretary & Director for the National Cyber Security Division

ENERGY delivery systems are critical to the effective and reliable operation of North America's energy infrastructure (electric power generation, oil, and natural gas production, transmission, and distribution systems) provides energy for our way of life. Today's highly reliable and flexible energy infrastructure is only possible because of the energy delivery systems' ability to provide timely information to system operators and automated control over a large, dispersed network of assets and components. This vast and distributed control requires energy delivery systems to communicate with thousands of nodes and devices across multiple domains, exposing energy systems and other dependent infrastructures to potential harm from accidental and malevolent cyber attacks.

Cybersecurity is a serious security challenge for the energy sector. Energy control systems are uniquely designed and operated to control real-time physical processes that deliver continuous and reliable power to support national and economic security. As such, they require security solutions that meet unique performance requirements, design, and operational needs. Cyber threats to energy delivery systems can impact national security, public safety, and our economy. Because the private sector owns and operates most of the energy sector's critical assets and the Federal government is tasked with national security, securing North America's energy delivery systems against cyber threats cannot be achieved by either the private or public sector working alone. Cybersecurity is a shared responsibility between the public and private sector.

The Department of Energy (DOE) is working to modernize the energy sector and integrate secure control systems. A common vision and framework for achieving this vision has been developed to guide the public-private partnerships that will secure energy delivery systems. This common vision, from the Roadmap to Secure Energy Delivery Systems, is that within ten years resilient energy delivery systems will be designed, installed, operated, and maintained to survive a cyber incident with no loss of critical function. The DOE Office of Electricity Delivery and Energy Reliability, Cybersecurity for Energy Delivery System (CEDS) Program, has implemented a multi-faceted program to address long-, mid-, and near term research, development, and implementation to meet the stringent cybersecurity requirements of the energy sector. The approach to addressing the cybersecurity needs of the energy sector that is being addressed through the CEDS Program and their public-private partnerships will be discussed.

Understanding Data Leak Prevention

Preeti Raman, Hilmi Güneş Kayacık, and Anil Somayaji

Abstract—**Data leaks involve the release of sensitive information to an untrusted third party, intentionally or otherwise. Many vendors currently offer data leak prevention products; surprisingly, however, there is very little academic research on this problem. In this paper, we attempt to motivate future work in this area through a review of the field and related research questions. Specifically, we define the data leak prevention problem, describe current approaches, and outline potential research directions in the field. As part of this discussion, we explore the idea that while intrusion detection techniques may be applicable to many aspects of the data leak prevention problem, the problem is distinct enough that it requires its own solutions.**

Index Terms—**Data Leak Prevention, Text Clustering Analysis, Social Network Analysis**

I. Introduction

ORGANIZATIONS increasingly may be harmed by data being revealed to unauthorized parties. Such data leaks can cause harm in a variety of ways. Improper handling of confidential data can violate government regulations, resulting in fines and other sanctions. Companies can be held liable for the release of customer and employee information such as credit cards and social security numbers. Further, loss of proprietary information to competitors can result in loss of sales and may even threaten the existence of an organization.

Data leak prevention (DLP) refers to products or techniques that attempt to mitigate some or all of these threats. DLP products are available from multiple vendors, including Symantec [1], CA Technologies [2], Trend Micro [3] and McAfee [4]. In contrast, data leak prevention has received little attention in the academic research community. This is not to say that DLP is a solved problem: indeed, current products are limited in what threats they address.

In this paper, we argue that data leak prevention is an area ripe for further research in that there are multiple hard problems of significant real-world interest that have not been rigorously studied. While DLP overlaps significantly with the field of intrusion detection, as we will explain that the overlap in potentially applicable techniques obscures the more significant differences in the respective problem requirements.

The rest of this paper proceeds as follows. First, we define the data leak prevention problem in Section II. Section III describes past related work in DLP. We describe the challenges of the problem in Section IV. Section V presents potential research directions in DLP. We address the issue of the overlap between DLP and intrusion detection and conclude in Section VI.

II. Data Leak Prevention Problem

There are numerous ways sensitive data can be revealed to untrusted third parties, as depicted in Figure 1. Thus, in order to discuss the data leak prevention problem, we investigate several factors including data repositories and available data leak channels. It is crucial to identify sensitive data repositories within an organization since selecting suitable prevention techniques naturally depends on the repository in question. Customer records, proprietary source code and sensitive documents on network shares are a few examples of repositories. Different prevention techniques may be appropriate for different data states: 1) at rest (at the repository); 2) in motion (over the network), and 3) in use (at the endpoint) [5]. When the data is at rest, the repository can be protected with access control and audit. However, when the data is in motion or in use, prevention using access control becomes increasingly difficult. For in motion and in use scenarios, the data leak prevention mechanism should be sufficiently context aware to infer the semantics of communication.

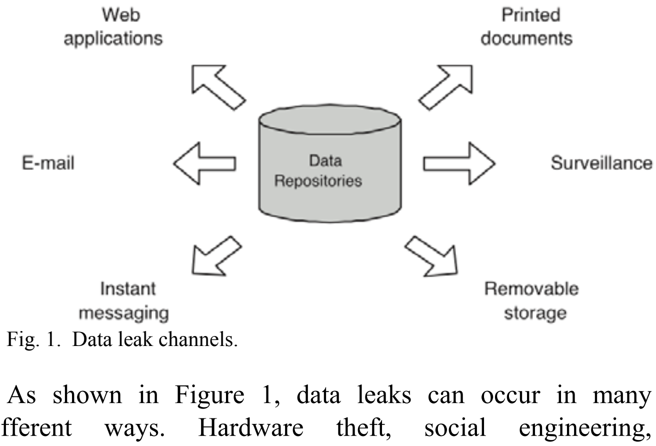


Fig. 1. Data leak channels.

As shown in Figure 1, data leaks can occur in many different ways. Hardware theft, social engineering, surveillance, and the mismanagement of printed documents are a few of the more traditional data leak channels. Additionally, electronic communications such as instant messaging, web applications and email provide additional

The authors gratefully acknowledge the support of CA Labs, MITACS-Accelerate, NSERC Discovery and NSERC ISSNet funding programs.

P. Raman, H. G. Kayacık, and A. Somayaji are in the School of Computer Sicnece, Carleton University, Ottawa, ON, K1S 5B6, Canada (email: {praman, kayacik, soma}@ccsl.carleton.ca}

challenges. These electronic channels are highly utilized in organizations and provide means to quickly and easily send data to a third party. While traditional data leaks can be more suitably defended with traditional approaches [6], lightweight and context aware techniques, which can infer who is communicating and what is being communicated, are needed to prevent data leaks in electronic communications.

III. Current Approaches

Various companies have recently started providing data leak prevention solutions. While some solutions secure 'data at rest' by restricting access to it and encrypting it, the state of the art relies on robust policies and pattern-matching algorithms for data leak detection. On the other hand, related academic work in data leak prevention focused on building policies [7], developing watermarking schemes [8], and identifying the forensic evidence for post-mortem analysis [9].

Vachharajani et al. [7] provides a user-level policy language for hardware-enforced policies, which ensures that the sensitive data does not reach untrusted output channels through network communications, files, and shared memory. The proposed runtime information flow security system assigns predefined labels to the data and policies are enforced at the hardware level to ensure the data flow complies with the policies. Needless to say, such an approach involves the labor-intensive task of the definition of labels, policies, and requires hardware that supports information flow security.

Lee et al. [9] approaches data leak prevention from a forensics point of view and identifies the set of files needed to detect data leaks on a Windows operating system. The authors argue that delaying the collection of forensic data will have detrimental effects in the effectiveness of a data leak prevention system; hence, they propose an efficient method to collect the basic information needed to detect data leaks by investigating five crucial system files: 1) the installation record file, 2) the system event log, 3) the windows registry, 4) the browser history, and 5) the core file in NTFS. Their approach is limited to file system-level data leaks on Windows platforms.

The synthetic decoy scheme of White et al. [8] focuses on the data leaks on large databases of personal records and proposes realistic decoy records to identify the source of data leaks, particularly when multiple databases are concerned. By creating uniquely identifiable, but semantically plausible personal records, the database can be digitally watermarked. Thus, any data leak from the database will contain the decoys unique to the database in question, hence, revealing the source of the leak. Such an approach, by nature, focuses on the postmortem identification of the data leak source not the real-time detection of the leak itself.

The current state-of-the-art in commercial data leak prevention focuses on pattern-matching, which suffers from the general shortcoming of misuse detection techniques; an expert needs to define the signatures. Given the elusive definition of data leaks, signatures should be defined per corporation basis, making the widespread deployment of current data leak prevention tools a challenge. On the other hand, the relevant academic work on data leak prevention and text mining takes a forensics approach and mainly focuses on post-mortem identification. Thus, detecting complex data leaks in real-time remains an understudied field.

IV. Challenges

A. Encryption

As discussed in Section II, different prevention mechanisms are needed to cover different states of data. In particular, detecting and preventing data leaks in transit are hampered due to encryption and the high volume of electronic communications. While encryption provides means to ensure the confidentiality, authenticity and integrity of the data, it also makes it difficult to identify the data leaks occurring over encrypted channels. Encrypted emails and file transfer protocols such as SFTP imply that complementary DLP mechanisms should be employed for greater coverage of leak channels. Employing data leak prevention at the endpoint – outside the encrypted channel – has the potential to detect the leaks before the communication is encrypted.

B. Access Control

Access control provides the first line of defense in DLP. However, it does not have the proper level of granularity and may be outdated. While access control is suitable for data at rest, it is difficult to implement for data in transit and in use. In other words, once the data is retrieved from the repository, it is difficult to enforce access control. Furthermore, access control systems are not always configured with the least privilege principle in mind. For example, if an access control system grants full access to all code repositories for all programmers, it will not effectively detect data leaks where a programmer accesses a project that he/she is not involved in.

C. Semantic Gap in DLP

DLP is a multifaceted problem. The definition of a data leak is likely to vary between organizations depending on the sensitive data to be protected, the degree of interaction between the users and the available communication channels. The current state-of-the-art, which is reviewed in Section III, mainly focuses on the use of misuse detection (signatures) and post-mortem analysis (forensics). The common shortcoming of such approaches is that they lack the semantics of the events being monitored. When a data leak is defined by the communicating parties as well as the data exchanged during the communication, a simple pattern matching or access control scheme cannot infer the nature of the communication. Therefore, data leak prevention mechanisms need to keep track of who, what and where to be able to defend against complex data leak scenarios.

TABLE I
A SUMMARY OF RELEVANT DATA LEAK PREVENTION MECHANISMS

	Data State	Data Channel	Detection based on	Objectives and Remarks
Pattern matching	In use	• E-communications	Database of data leak signatures	• Develop misuse signatures. • If signature match occurs, indicates data leak. • Attack mutations/modifications are hard to handle.
Access control	At rest	• Databases • Repositories	Access control list	• Control the access of a resource. • Grant access if user is on the "white-list." • A perimeter defense, hence does not work proactively.
Text clustering	In transit In use	• E-communications • Repositories (focused on text)	A set of clusters with semantic meaning of he communications	• Identify the nature/topic of communication. • Unusual activity requires attention. • Needs to be scalable and results should be easy to comprehend.
Social network analysis	In transit In use	• E-communications • Repositories (focused on user interaction)	Social network graph of users	• Discover social networks of collaboration. • Drastic changes in social networks require attention. • Social networks need validation before use.

A. Collaboration

In order to be able to identify the 'outsider' in a communication, the collaborating parties should be identified. However, identifying collaborators is not a straightforward task. While a naive metadata approach can consider using access control mechanisms (e.g. to determine the programmers, managers, administrators, etc.) such an approach is not sufficient to capture heterogeneous groups where people can belong to more than one group. Furthermore, the temporal nature of collaborations should be addressed. As time passes, new collaborations are formed and existing ones disappear. Thus, the analysis of collaborations should not be regarded as a one-time task but as a continuous task to be carried out on regular intervals.

V. FUTURE DIRECTIONS

As summarized in Table I, the biggest shortcoming of the state-of-the-art and the relevant previous work is that they attempt to detect data leaks without an understanding of the communication context. However, the complex data leaks are in semantics (i.e. the content of the conversation) not in syntax (i.e. whether a pattern resembling social insurance numbers occurs). Thus, in order to address the semantic gap problem in data leak prevention, new research directions should be explored to provide the semantic summarization of communications. The main focus is identifying in-transit and in-use data leaks. In this section, we review the text clustering and social network analysis approaches that are likely to aid in building context aware DLP solutions.

A. Text Clustering

Text clustering [10] is an exploratory data analysis technique that aims to identify the natural groupings (i.e. 'clusters') within a text corpus. Each cluster contains similar documents, according to a similarity metric such as Euclidean distance. From a data leak prevention perspective, text can be collected from numerous sources, an example of which is email. The clusters of text can serve as equivalence classes (content summaries) that can then be labeled to provide semantic meaning. Thus, by applying clustering to email communications, it is possible to infer the subject of the communication in a privacy-preserving manner. Based on the subjects about which a user communicates, a deviation from the 'usual' is flagged and further analyzed for data leaks.

Text clustering, which places documents with similar properties within the same group, has been utilized for summarizing large corpus of documents. Cavnar et al. [11] employed n-gram representation of text-for-text categorization. The documents are represented as n-grams, in which an n-gram is an n character slice of a longer string. Taking advantage of the Zipf's Law [12] in human language text, they identified the language of the text based on the most frequent 300 n-grams. Furthermore, they demonstrated that, the n-grams below 300 are specific to text topic, hence providing a means to cluster the text according to context.

In terms of the analysis of email as a text corpus, Chow et al. [13] aimed to detect the inferences in sensitive documents by applying various data mining algorithms to the Enron email corpus [17], which contains the email communications of top-level Enron employees before and during the Enron scandal. The inferences are determined based on co-occurrence of terms in the text corpus. Similarly, Keila et al. [14] proposed a method for detecting deceptive emails, based on the expectation that people use fewer first person pronouns and more negative emotion and action verbs. Singular value decomposition is utilized to visualize email messages and identify the outliers, which correspond to deceptive emails. The previous relevant text mining approaches [11][13][14] focus on document summarization in general, without a data leak prevention focus.

Applying text clustering to data leak prevention involves monitoring corporate email communications for a period of time to identify the clusters of topics, in other words, communication subjects. The output of clustering may be difficult for a human to comprehend without further processing such as in the case of the commonly utilized k-means clustering, which represents centroids (i.e. cluster centers) as high-dimensional vectors. However, clustering algorithms such as approximate divisive hierarchical clustering [10][15] can provide a cluster-identifying tree, which the administrator can analyze and modify, if necessary. Thus the resulting visualization can be utilized to assign semantic meaning to the clusters manually or automatically. During deployment, when an email communications is processed, the most similar cluster is employed to assign the topic of the email. If there exists a substantial deviation of communication pattern (in terms of the context, frequency and the involved parties), the resulting communication is flagged for further analysis.

B. Social Network Analysis

Social network analysis involves the mapping and measuring of relationships between people, groups, and organizations by representing the relationships in terms of nodes and connections. Social networks can be derived from communication channels such as email, forum discussions, and social networking sites. Analysis of social networks can improve our understanding of the relationships and groupings between the parties involved in electronic communications, email in particular. Thus, the goal of social network analysis for data leak prevention is to identify the communication patterns within the organization and employ feedback from the administrator to identify unusual communications to uncover to data leaks.

Diesner et al. [16] performed a social network analysis of the Enron emails. The social networks extracted from the email communications take the form of directed graphs where each edge is weighted according to the cumulative frequency of emails exchanged between the nodes (i.e. people) in the graph. The comparison of the communication structure before and during the crisis indicated a movement toward communicating only between trusted parties, due to accountability. Furthermore, immediately after the bankruptcy became public, an increase in outward communications is observed potentially a likely outcome of people seeking more information on the recent events [16].

Applying social network analysis in data leak prevention involves monitoring online collaboration (email, document and code repositories) to discover the social networks of collaboration. The discovered social networks are vital in identifying collaborators such as a team of developers working on the same code repository or a group of employees exchanging emails to perform a task (e.g. preparing for a meeting). Social network analysis has the potential to discover collaborations that are not documented as a part of company policy or access control. Proper visualization of social networks can be presented to the administrator for manual or automatic validation. During deployment, if a substantial change in the social network is observed, it is flagged for further analysis since it can reveal: 1) a dissolving social network, 2) a merging social network, or 3) inclusion of an untrusted party, which is potentially a data leak.

VI. Conclusion

DLP is a multifaceted problem. Determining the sensitive data to be protected, identifying the legitimate use of the data and anticipating data leak channels require knowledge of the internal business logic of the corporation. Thus, there is no one-size-fits-all solution. In addition to traditional data leak channels such as hardware theft, the widespread use of electronic communications such as email makes it easy to leak sensitive data in a matter of seconds. Both data leak prevention and intrusion detection share the same common goal, which is to detect potentially harmful activity. Thus, the commercial approach typically employs similar techniques to solve data leak prevention. However, data leak prevention focuses on what (is leaked) as opposed to intrusion detection, which focuses on who (is breaking in). DLP is a complex problem, in which the threat usually originates from the 'inside' and the definition of 'misuse' is elusive. Data leaks can occur by accident between individuals who are completely legitimate. The detection of such data leaks requires an understanding of semantics. The current state-of-the-art in data leak prevention mainly utilizes misuse detection to detect data leaks, where a signature acts as a data leak description. However, misuse detection cannot scale well in data leak prevention since the data leak signatures – highly dependent on the internal business logic – should be developed per organization to minimize false positives and maximize detection rate. Furthermore, misuse detection does not possess the sufficient context awareness to detect complex data leak scenarios, where the data leak is in the semantics, not in syntax.

In this paper, we reviewed the current state-of-the-art as well as potential research areas which can provide context-aware data leak prevention solutions, as summarized in Table I. Text clustering and social network analysis discussed in Section V focus on summarizing the electronic communications in a lightweight privacy-preserving manner and inferring the semantic meaning. This allows data leak prevention to go beyond pattern-matching and detect complex data leaks based on who is involved in the communication well as what information is being exchanged.

Acknowledgment

The authors would like to thank Carrie Gates from CA Labs for the valuable feedback.

References

[1] Machine Learning Sets New Standard for Data Loss Prevention: Describe, Fingerprint, Learn [White Paper]. (2010, December 14). *Symantec*. [Online]. Available: http://eval.symantec.com/mktginfo/enterprise/whitepapers/b-dlp machine learning.WP en-us.pdf

[2] CA DLP: information protection and control [Product Sheet]. (2010). *CA Technologies* [Online]. Available: http://www.ca.com/~/media/Files/productbriefs/dlp-12-5-ps.pdf

[3] Trend micro data protection: Solutions for privacy, disclosure and encryption [White Paper]. *Trend Micro* [Online]. Available: http://us.trendmicro.com/.../datalossprevention/wp02_dlp-compliance-solutions_100225us.pdf

[4] McAfee host data loss prevention [Data Sheet]. *McAfee* [Online]. Available: http://www.mcafee.com/us/resources/data-sheets/ds-host-data-loss-prevention.pdf

[5] Data leak prevention [White Paper]. (2010, September 14). *Information Systems Audit and Control Association (ISACA)* [Online]. Available: http://www.isaca.org/Knowledge-Center/Research/Documents/DLP-WP-14Sept2010-Research.pdf

[6] J. Livingston, "Tips and Strategies to Protect Laptops and the Sensitive Data They Contain," *Information Systems Audit and Control Association (ISACA) Journal,* vol. 5, pp. 1-3, 2007.

[7] N. Vachharajani, M. J. Bridges, J. Chang, R. Rangan, G. Ottoni, J. A. Blome, G. A. Reis, M. Vachharajani, and D. I. August, "Rifle: An architectural framework for user-centric information-flow security," in Proceedings of the 37th annual IEEE/ACM International Symposium on Microarchitecture (MICRO 37), Portland, OR, USA, 2004, pp. 243–254.

[8] J. White, and D. Thompson, "Using synthetic decoys to digitally watermark personally-identifying data and to promote data security," in Proceedings of the International Conference on Security and Management (SAM 2006), June 2006, pp. 91–99.

[9] S. Lee, K. Lee, A. Savoldi, and S. Lee, "Data leak analysis in a corporate environment," in Proceedings of the 2009 Fourth International Conference on Innovative Computing, Information and Control (ICICIC '09), Las Vegas, NV, June 2009, pp. 38–43.

[10] J. Han. *Data Mining: Concepts and Techniques.* San Francisco, CA, USA: Morgan Kaufmann Publishers, Inc., 2005.

[11] W. B. Cavnar, and J. M. Trenkle, "N-gram-based text categorization," In *Proceedings of 3rd Annual Symposium on Document Analysis and Information Retrieval (SDAIR-94),* 1994, pp. 161–175.

[12] G. K. Zipf, *Human Behavior and the Principle of Least-Effort.* Cambridge, MA: Addison-Wesley, 1949.

[13] R. Chow, P. Golle, and J. Staddon, "Detecting privacy leaks using corpus-based association rules," in *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08).* Las Vegas, NV, USA, August 2008, pp. 893–901.

[14] P. S. Keila, and D. B. Skillicorn, "Detecting unusual email communication," in *Proceedings of the 2005 conference of the Centre for Advanced Studies on Collaborative research (CASCON '05),* IBM, October 2005, pp. 117–125.

[15] H. Inoue, D. Jansens, A. Hijazi, and A. Somayaji, "Netadhict: a tool for understanding network traffic," in *Proceedings of the 21st conference on Large Installation System Administration Conference (LISA'07),* Dallas, TX, USA, November 2007, pp. 1–9.

[16] J. Diesner, T. L. Frantz, and K. M. Carley, "Communication networks from the Enron email corpus: It's always about the people. Enron is no different," *Computational & Mathematical Organization Theory* [Online], vol. 11, October 2005, pp. 201–228. Available: http: //portal.acm.org/citation.cfm?id=1110938.1110942

[17] J. Shetty, and J. Adibi, "The Enron email dataset database schema and brief statistical report," Information Sciences Institute, University of Southern California, Los Angeles, CA, USA, Technical Report, 2004.

Breaching & Protecting an Anonymizing Network System

Jason W. Clark, and Angelos Stavrou

Abstract—**The protection of personally identifiable information (PII) is paramount for certain organizations. As a result, these organizations are currently employing both commercial products and open source Anonymizing Network Systems (ANS) as part of their standard network defenses to anonymize their outbound traffic. In this paper, we analyze the benefits and caveats of a typical deployment of an ANS. We call this the Case-Study Anonymizing Network System (CSANS) and we show that while network-level anonymity systems are better at protecting end-user privacy than having no ANS in place, they are unable to thwart de-anonymization attacks aimed at applications and private data of end-users. Indeed, the anonymity protection that ANS users enjoy can be bypassed by targeted attacks against both applications and users. To further explain our claims and quantify the different types of attacks and intruder capabilities, we introduce two threat models. We demonstrate and substantiate our claims using a targeted experiment against actual operational scenarios of users who are relying on the CSANS in a real organization to complete their research tasks. To that end, we set up an experiment to guide users (often referred to as researchers) to a website containing both non-malicious and malicious files. The downloading and opening of these files can enable an intruder to expose the true identities of the users even though the users were using a protected, isolated ANS connected machine. The main takeaway from this study was that both the administrators and CSANS users were not educated about quite prevalent attack vectors for compromising client systems and violating user privacy. Finally, we discuss the danger of using location-based services while on an ANS and we show that network-level anonymizers are insufficient. We conclude with recommendations for defending against anonymity attacks that aim for the application-level.**

Index Terms—**Anonymizing Network Systems, Security, Privacy, User and Application Attacks, User Training and Awareness, Traffic Analysis, Defenses and Countermeasures.**

I. Introduction

THE Internet offers a low-cost, low-risk, and high-value of return intelligence gathering and archival system. With the advent of social networking and information aggregation, the Internet has become an information highway storing copious amounts of information that is typically unintended for the intruder, but is freely available and relatively easy to discover for an analyst.

Often users of computer systems perform operations on the Internet, such as searching for information, which they would like to keep anonymous. One popular way to help protect a user's identity against a possible intruder is to implement an anonymizing network system (ANS). The field of anonymous communications and specifically, ANS started in 1981 with David Chaum's Mix-net [3]. In general, an ANS is a system or set of systems that is designed to protect a user's identity while they are using a computer system that can access the network. Therefore, an ANS is a tool that attempts to make activity on the Internet untraceable.

As described by Pang, et al., the first goal of an ANS is aimed at preventing the true identities of specific hosts from being leaked such that an audit trail of user activity cannot be formed. The second goal is to prevent the true identities of internal hosts from being leaked such that a map of supported services can be constructed. The third goal is to prevent the leakage of specific security practices within the publishing organizations network [28].

When implemented correctly, an ANS will access the Internet on the user's behalf and in theory protect personally identifiable information (PII) by hiding the source computers identifying information along the way. The goals of any ANS is to first hide the structural information about the network on which traces are collected and second to prevent the assembly of behavioral profiles for users on that network, such as the sites that they browse [1]. The ultimate goal of Internet anonymization is to allow a host to communicate with a non-participating server in such a manner that nobody can determine the user's identity.

Simply implementing an ANS will not necessarily protect a user's identity. However, it is safe to say that ANS are better than no system at all. As we will describe in a later section, a user will have to be careful and cognizant of what sites they visit and what actions they perform even when "protected" by an ANS. Often users of ANS gain a false sense of security as they are under the impression that they are completely protected and anonymous. As we will show, this assumption could not be further from the truth.

There exists a plethora of different ANS including the popular Anonymizer [1] and Tor [2] that are available on the market today. Other ANS solutions exist and they include but are not limited to: Browzar [20], JAP [21], Privoxy [33], and SafeSurf [22]. As a result, these ANS are predominantly built

Jason W. Clark and Angelos Stavrou are with the Computer Science Department in the School of Information Technology and Engineering at George Mason University, Fairfax, VA 22030 USA (e-mails: {jclarks, astavrou}@gmu.edu).

with the goal of protecting the network layer of the OSI model in mind. While this does provide a certain level of anonymity and identity protection compared to no system at all, it does leave itself vulnerable to a variety of different styles of attacks.

The focus of this paper is the analysis of the actual anonymity that an ANS provides. We assume that this system is going to be used to protect the identity of a user or an organization in a real-world setting using operational scenarios. The actual setup employed a widely used variation of the ANS offered by Anonymizer, Inc. To gather experimental evidence, we developed a case-study based on a typical ANS deployment in a large US-based organization. We call this version the case-study anonymizing network system or CSANS. The CSANS represents a typical implementation of an ANS that is available today. The inner-workings of the CSANS is described in detail in section III B.

The purpose of this study is to introduce the shortcomings of the CSANS by conducting attacks that focus on the actions of the users. We were approached by the management team at the case-study agency about the security and privacy of the CSANS. Specifically, the CSANS management was interested in determining whether the users were performing actions that could lead to their identity being discerned. The goal of this study is to allow defenses to be developed that can help reduce the likelihood that attacks against ANS are successful.

The main argument that we make is that organizations and users can't solely rely on an ANS to protect their identity. Rather we warn users to be cognizant of their actions when on an ANS and provide users recommendations and defenses to help protect against the intruder attack vectors we outline.

This study is significant because it shows that users behind an ANS are not as protected as they need to be. By completing this study, we show which attacks are the most successful and more importantly why they are successful.

The concept of researching ANS and their ability to protect a client's identity is something that has been studied for the last decade. Also, the concept of basic attacks by analyzing web traffic and attempting to perform application-layer attacks against systems and users is not original either. However, the merging of the two concepts is a new idea.

This paper focuses on explaining how the security of ANS such as Anonymizer, Tor, and CSANS can be "broken" by performing a variety of different and often unsophisticated attacks. It is this new outlook that makes the research original and worthwhile in doing.

Furthermore, our research is original in the sense that it is a hard look at all the essential ingredients of a specific attack used against a real-world ANS. We will conclude the research by offering recommendations to users of ANSs as to best defend against the attacks that we outline.

While it is true that we are predominantly focusing on CSANS, our research will be able to scale to many other ANSs. The reason for this is because they all have the potential to suffer from the same kinds of attacks that we describe.

The main contributions of this paper are we:

1) Introduce the concept of ANS and the current research literature in the field of ANS attacks
2) Examine how real users behave while on a real-world implementation of an ANS
3) Show that unsophisticated attacks can be quite useful for an intruder
4) Prove that users and a lack of education remain the weakest link when trying to keep privacy intact

II. Setup

This paper considers three main methods that could be used by an intruder to possibly discern the true identity of the user. The first method of potentially de-anonymizing a user is by a user clicking on a file that has logging capabilities enabled. This would allow the intruder to perform traffic analysis and find a correlation between the anonymized IP traffic and the user's real IP traffic.

Next, the intruder could create a malicious file that, when opened with a vulnerable application on a vulnerable operating system, could exploit the user giving the intruder the ability to take control of the system.

The last method we consider is attributable searching while on a non-attributable ANS. We define attributable searching to be the action of performing search queries that could yield information that might potentially de-anonymize the user. The intruder could gather attributable information based on search queries including, but not limited to, local weather, sports, and restaurants. Figure 1 shows the different de-anonymization methodologies discussed in this paper.

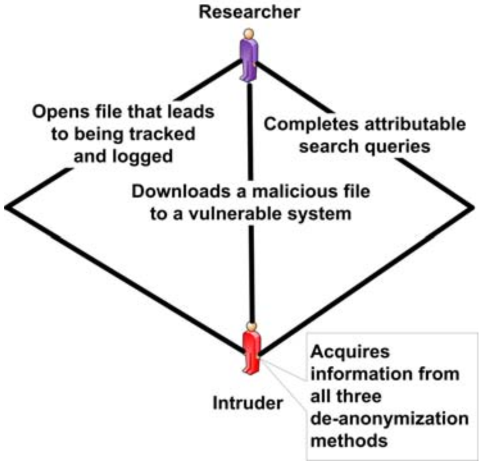

Researcher
Opens file that leads to being tracked and logged
Completes attributable search queries
Downloads a malicious file to a vulnerable system
Intruder
Acquires information from all three de-anonymization methods

Fig. 1. Possible de-anonymization methods.

The experiments conducted required the research manager of the case study organization to design a fictitious research task. The research task was designed entirely by the research manager with no assistance from the authors of this paper. Therefore, we conducted the experiment on a real operational environment and gathered the reported results from field experiments using real human subjects.

Figure 2 shows the relationships between the key players and the entities associated with the experiment.

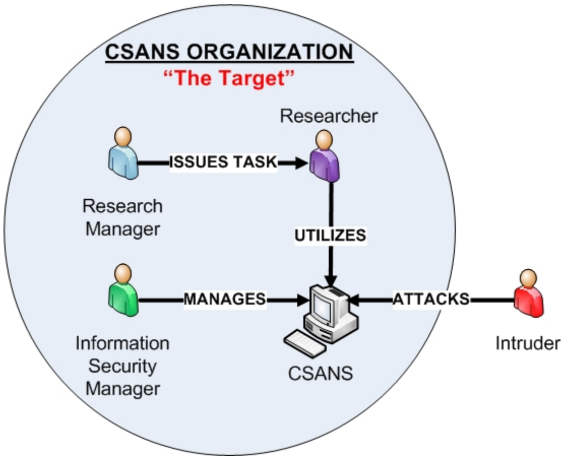


Fig. 2. Relationship between key players in the experiment.

As depicted in Figure 2, the research manager issues a specific task to the researcher which will require the use of the CSANS to accomplish. The information security manager is solely responsible for developing, managing, monitoring, and securing the CSANS. The intruder is assumed to be someone outside the organization who is interested in breaking the anonymization of the CSANS organization and its users in an effort to gain the researcher's PII. While it is possible that the intruder is an inside threat, we assume that the intruder is not associated with the CSANS organization. Figure 3 shows the interactions between the key players and the entities associated with the experiment.

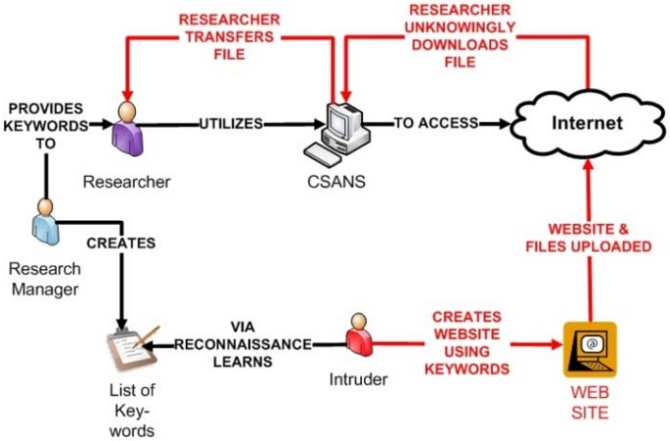


Fig. 3. Interactions between key players and entities in the experiment.

A. Ethics and Institutional Review Board for Human Subjects

Given the sensitivity and potential ethical issues associated with this type of experiment, some organizations and its users may feel that purposely trying to compromise systems and discover PII is a violation. However, we were authorized to conduct this experiment against the CSANS users with the support of the internal review board and permission of upper management.

To address any ethical issues that could stem from a study that involves humans, we requested the permission of the organization's review board, who subsequently authorized this experiment to ensure that the CSANS and its users were following best practices when it came to protecting the anonymity of the organization.

One of the major difficulties we encountered was to prevent people not associated with the experiment from accessing our website and downloading malicious files on to their system. We considered putting up a firewall to block access to our website except from a certain range of IP addresses. This proved to be difficult as the CSANS IP address changes randomly on a daily basis as part of the anonymization process.

Next, we considered password protecting our website, but were concerned about negatively impacting the experiment. This is because it would not have made sense for the researchers to know the password associated with our experimental website. However, we felt that this was the best option and worth preventing the risk of adversely infecting unsuspecting users.

Specifically, a password was used to protect the malicious files listed in Table III and the password was given to the research manager. The research manager then created a fictitious story as to how he learned the password and passed it on to his researchers as part of the task. While not a perfect solution, we did attempt to prevent people who accidentally visited our site from infecting their machines. We leave it to future experiments and research to find a more elegant solution to this problem.

B. Roles and Responsibilities

In Table I, we describe the roles and responsibilities of several key people who participated in the experiment.

TABLE I
ROLES AND RESPONSIBILITIES

Role	Responsibilities
Research Manager	Authorizes experiment Creates and provides list of keywords to researchers Manages researchers Authorized to stop experiment
Information Security Manager	Develops, secures & monitors CSANS Assists with security-related issues
Paper Authors	Act as intruder Develop experiment Conduct experiment Create website and corresponding files Capture and analyzes data Document and report on findings of experiment
CSANS Users (Researchers)	Do not have knowledge of experiment Complete research task independently with no collaboration
CSANS Organization	Management has knowledge of research task or experiment whereas user community does not

III. Background Information

In this section, we briefly discuss anonymity on the Internet, inner-workings of the CSANS, targeting and profiling, current literature, Google location, and an introduction to our intruder threat models.

A. Anonymity on the Internet

Depending on the individual reason for accessing the Internet, anonymity may or may not be an important issue. For example, anonymity requirements may not be a concern for someone who is using the Internet to access a news site. Similarly, there is a growing trend for people willing to disclose their identity by posting personal information about themselves on a social networking site.

Conversely, users in defense and intelligence-related fields almost always have a requirement to remain anonymous when researching and compiling data from other (hostile) countries; if their identity is known, it makes completing the research far more difficult and possibly even dangerous. For example, the groups and countries that they are researching could potentially retaliate in response to the research being conducted without their consent. For the purposes of this paper, the CSANS organization consisted of real users who needed to remain anonymous in order to complete the research tasks that they were given.

B. Inner-workings of the CSANS

The CSANS is a variation of Anonymizer offered by Anonymizer, Inc [1]. The CSANS has been designed to prevent an open Internet connection from being exploited. All traffic is routed through dedicated hardware housed in a secure facility. Only authorized administrators are given access to this facility. The CSANS maintains a large pool of IP addresses and is able to easily rotate and replace IP addresses that have possibly been compromised. The users of CSANS should expect to see their IP address change about once every 24 hours. This concept is known as the IP rotator scheme.

In this scheme, anonymized users are mixed with “regular” worldwide consumers. The behaviors of both types of users (anonymized and regular) are mixed together in an effort to prevent behavioral patterns from being noticed. The regular users are diverting attention away from the anonymized users by virtue of the search queries and traffic that they are generating.

The hardware associated with CSANS is an enhanced version of the Juniper Virtual Private Network (VPN) box. The CSANS includes two interfaces: 1) the front-end and 2) the back-end. The front-end interface has a specific network IP address, and the back-end has a completely different network IP address. For security purposes, the IP addresses are visible internally, but not externally. The firewall routing for CSANS is designed to protect users from accidentally visiting sites associated with the specific organization that implemented CSANS. The routing rules are set to essentially mask the source and destination as to provide a second layer of anonymity.

The CSANS is hosted on a Windows server virtual machine, and users of the CSANS have to remote desktop into the system to get access to CSANS as seen in Figure 4.

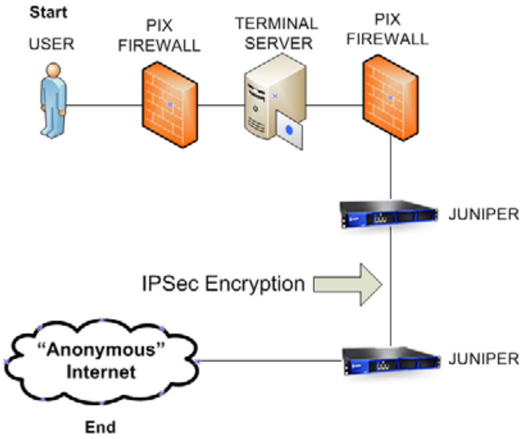


Fig. 4. CSANS Architecture.

The CSANS environment is locked down so that only certain users have administrative privileges. Since this particular implementation is a shared environment, all user requests including software installations need to be authorized. This follows best practices for the security aspects of the server. Specifically, in an effort to keep the CSANS clean from malicious code, the system administrators incorporate the use of a virtual machine snapshot. Once a week Windows updates, anti-virus updates, and software installations are performed resulting in a new snapshot being taken. If the current snapshot becomes corrupted or infected with malware, CSANS would be able to revert back to a known good snapshot.

One of the key features of CSANS is the ability to transfer files from CSANS back to the client OS. While this is extremely helpful to the end-users in terms of usability and functionality, it is also a possible security concern. The rationale for implementing the transfer file component is because the CSANS users have a requirement to share files and to create and analyze reports using software from their computers.

The developers of the CSANS implemented a simple file sharing system. The file sharing system utilizes a FTP client on their machines along with a FTP server that could run from the CSANS environment. The CSANS users simply save their files to a “Home” directory and using FTP moved the files to their client machine.

C. Targeting and Profiling

In this section, we discuss targeting a specific organization and creating a document to be used in the attack. The goal of the intruder is to get the targeted users to download the malicious document. To do this, the intruder needs to profile the organization so the intruder can create documents that are appealing. Ideally, the documents that are uploaded on the website will be related to a topic that is of interest to the user as shown in Table III. In addition, most users feel comfortable and safe opening Adobe PDF and Microsoft Word files.

Therefore, if the intruder can customize the document to make it look like it was an organizational newsletter, job posting, company form, etc., it is more likely that the document will be downloaded [5]. If intruders can build up a good data set, it would only be a matter of time before a user comes to the intruder's website and downloads something malicious.

The attack that we will demonstrate is based on the idea that the intruder can use "common and public" knowledge about a particular organization and its users. If, for instance, the intruder can infer and attract attention about the target, it will be more likely that the attack will succeed. Through profiling, an intruder could potentially identify the CSANS organization and their research tasks and create documents that are related to these same areas of interest.

D. *Survey of Current Research on ANS Attacks*

Almost all ANSs, including the one used for this case study, are vulnerable to a variety of different types of attacks. In this section, we describe current research that is being conducted to attack ANSs. The literature review considers traffic analysis, application, and network-based ANS attacks. In addition to what was previously discussed, there is another significant body of work in the field of traffic generated by real-time ANSs.

Schear and Nicol [6] discuss the idea that network encryption, both at the packet and session layer, is used widely for securing private data. They use simulation and an analytical model to examine the impact on user experience via a scheme that masks the behavior of real traffic by embedding it in synthetic and encrypted cover traffic.

Coull, et al. [4] attempt to solve the problem of publishing data that can potentially leak sensitive information about the publishing organization. The article introduces the problem by suggesting that it is imperative that trace and log data be made publicly available for verification and comparison of results. The authors attempt to conduct an analysis and create techniques to infer sensitive information from these network traces. The study, as performed by the authors, demonstrates that there are more substantial forms of information leakage that inherently compromise current anonymization methodologies.

Wang, Chen, and Jajodia [9] describe the concept of watermarking network traffic in order to break the ANS. The watermarking aspect in this case can potentially allow an intruder to influence the traffic thus allowing them to discern the identity of the user. The authors described how they were able to successfully penetrate the Anonymizer, Inc. "Total Net Shield", the "ultimate solution in online identity protection" [9], which is almost the same ANS that is used in our case study.

Coull et al. [10] improve on previous research done on reconstructing web browsing activities from anonymized packet-level traces. This is accomplished by accounting for real-world challenges such as browser caching and session parsing. This research evaluated the effectiveness of the author's techniques by identifying the front pages of the fifty most popular websites on the Internet.

Hopper, Vasserman and Chan-Tin [11] present two attacks on low-latency ANS schemes. The first attack allows a pair of colluding websites to predict, based on local timing information and with no additional resources, whether two connections from the same Tor exit node are using the same circuit with high confidence. The second attack requires more resources, but allows a malicious website to gain several bits of information about a client each time a user visits the site. The authors evaluate both of their attacks against the Tor network and the MultiProxy proxy aggregator service.

Casado and Freedman [12] state that online services often use IP addresses as client identifiers when enforcing access control decisions. The idea of the paper is to determine the impact that edge technologies such as NAT, proxies, and DHCP have on the utility of using IP addresses as client identifiers. Specifically, whether the IP address of an incoming client is a useful identifier of access control decisions and to explore the extent to which edge opacity obscures a server's view of the client.

Coull et al.'s [13] primary concern is to evaluate the efficacy of network data anonymization techniques with respect to the privacy that they afford. Specifically, the authors are trying to make the network flow uniquely identifiable even after it has gone through the ANS. They are also considering techniques for evaluating the anonymity of network data and to simulate behavior of an intruder whose goal is to de-anonymize the objects (host or web pages).

Jung, et al. [14] described the design and implementation of a system called Privacy Oracle that is capable of finding application leaks. The authors attempt to solve the problem of application leaks by using black box differential testing. The creators of Privacy Oracle are most interested in what information was leaked, when it is exposed, and who can receive it.

Scott and Sharp [15] use new tools and techniques which address the problem of application web security. The authors describe their solution to address the problem of application layer web security by describing a scalable structuring mechanism facilitating the abstraction of security polices from web applications developed in heterogeneous multi-platform environment. The second aspect to their solution is to represent a tool which assists programmers developing secure applications which are resilient to a wide range of common attacks. Finally, the third aspect is to report results and experiences arising from the implementation of these techniques.

Huber, Mulazzani, and Weippl [29] completed a full-analysis of HTTP usage and information leakage for the Tor network. The authors performed an analysis of the web browsing behaviors of Tor users to show which websites were of interest to them. Next, they determined an upper bound on how vulnerable Tor users were to sophisticated de-anonymization attacks.

Kadloor, et al. [30] were able to show that a remote intruder can obtain significant traffic timing and volume information about a particular user simply by observing the round-trip time of the pings.

Wandracek, et al. [31] introduce a novel de-anonymization attack that exploits group membership information that is available on social networking sites. This paper illustrates the tactic of using a malicious website to launch a de-anonymization attack to learn the identity of the visitors. The advantages of this type of attack are that it has a low cost and has the ability to affect a plethora of users.

Chaabane, Manils, and Kaafar [32] analyze Tor in the wild by setting several exit notes and distributing them worldwide. This is done to show that Tor is actually being misused as most observed traffic belongs to P2P applications.

Abbott, et al. [34] completely detail and describe a new attack on the anonymity of web browsing with Tor. The attack focuses on the user's web browser and works by sending a distinctive signal over the Tor network that can be detected using traffic analysis. Similar to the attacks we demonstrate, both the attack and the traffic analysis can be done by an intruder with limited resources.

Based on a review of current literature, we found that many security researchers are studying attacks that focus on traffic analysis, network-layer based attacks, and some application-layer based attacks. Furthermore, the focus of this research seems to be with Tor and not Anonymizer.

We introduced several different anonymization schemes and concepts including ANS, network trace anonymization, and to a lesser extent, proxy systems. While each of these systems have unique properties and disjoint security goals, we felt compelled to provide an overview of the current state of attacks against ANS. We introduced de-anonymization attacks that are complex and detailed in nature when compared to the attack that we are demonstrating. What we are suggesting is users and certain applications, if successfully compromised, can expose the organization, its users, and its tasks even if an ANS is implemented.

In fact, ANS are not designed to prevent basic attacks against applications and the users. The attacks we describe are quite effective in terms of how they can be used to produce user and computer information such as IP addresses, email addresses, source location, timestamps, and other pertinent system information.

.Location-based Exposure (Google Location)

While not an attack by definition, the Google Location feature has a major impact on the security and privacy associated with the users of Google. Specifically, the Google Location technology in certain cases identifies the user's true location. Google has massive amounts of technology and tools to be able to analyze traffic and find correlations between the user's search criteria and the user's likely location(s).

We define the concept of location-based exposure to be when users who pass through an ANS search for local: businesses, politics, weather, sports, etc. Essentially, users forget or are unaware that they are performing an attributable search that should not be done while on an ANS. As a result of this location-based exposure, Google is often able to correlate these searches and show the user's real location regardless of whether they are behind an ANS or not.

For example, suppose that the anonymized IP address of a given user is 123.45.67.89 and it shows that the location is anonymized to be in Ghana. However, multiple ANS users begin to search for Denver, CO weather conditions and the score from last night's Denver Nuggets basketball game. Google is able to identify this behavior and, given enough searches and traffic, will assume with a high likelihood that this IP address is not associated with Ghana, but instead with that of Denver, CO. The user's behavior is being used by Google to find the most likely location from which this user and corresponding IP address is coming. Given enough of this data, the Google Locator actually becomes quite accurate.

This problem is not necessarily specific to Google; although they are by far the biggest actor in developing this technology. Note that Google is not attacking in an effort to be malicious or uncover the true identity of any organization. Instead, they are simply trying to be more precise with their search results and advertisements. From Google's perspective, they want to be both complete and accurate with their search results.

This particular attack vector would almost certainly be useful in determining whether or not the user was behind an ANS. By the very nature of location-based exposure, the browsing history would be known as illustrated in the above Denver search query example. It might be difficult for an intruder to determine with certainty with what organization the user belongs. However, they may be able to narrow it down by geographic location, thus potentially leading to the CSANS user's physical location.

Intruder Threat Model

Threat modeling is a method of assessing and documenting the security risks associated with an application. The threat model for CSANS is based on the process outlined in Swiderski and Snyder [7].

First, we examine what PII the intruder seeks. The intruder is most interested in the true identity of a user behind the CSANS. Such information could include:

- User's real name
- User's place of employment (organization name)
- Real IP address (non-anonymized IP address)
- Browser configuration and browser history
- Sensitive documents that reside on users system
- Software and programs currently installed on CSANS
- Configuration and implementation of CSANS

One method for identifying and categorizing threats is known as STRIDE. This method is a classification of the effects of realizing a threat and it stands for Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege [7]. The STRIDE classification for various vulnerabilities is listed in Table II. Next, we consider internal and external threats. The following are a few threats associated with the CSANS.

- The CSANS users have been known to post sensitive documentation to fake email accounts for the purposes of storing data for future use.
- The CSANS administrator has full control over all the user accounts and passwords.
- The CSANS system has directories of sensitive documents that are stored on the server itself (before it is transferred). By logging into the server, an intruder could view these documents.
- The CSANS has a built-in transfer system that is designed for convenience rather than security.
- The anonymization is being done by a third-party.
- Users have been known to perform attributable search queries while on the CSANS.

In addition, we included a DREAD rating for each of the CSANS vulnerabilities that we identify in Table II. The DREAD rating is a method for characterizing the risk associated with vulnerabilities. The DREAD rating comprises Damage potential, Reproducibility, Exploitability, Affected users, and Discoverability [7].

The DREAD rating for the vulnerabilities associated with CSANS uses a three-point rating scale. A rating of one indicates that the impact is minor, the number of systems impacted is a small percentage, and the likelihood that the attack is identified is low.

In general, the lower score equates to minor damage, less reproducibility, less exploitability, fewer numbers of affected users, and less likely to be discovered. Conversely, a rating of three equates to major damage, more reproducibility, more exploitability, larger numbers of affected users, and greater likelihood for the attack to be discovered. The final DREAD rating score is simply the average of the scores for each of the five aforementioned categories.

Table II displays a non-exhaustive list of several vulnerabilities that the CSANS intruder could take advantage of along with the corresponding DREAD ratings.

IV. Methodology

The following sections describe our experimental methodology. Specifically, we discuss the experimental goals and objectives, experimental setup, and the creation of a fictitious research task and website.

A. Experimental Goals and Objectives

The primary goal of the experiment was to determine the extent to which we could use application and user-based attacks to discern the true identities of the CSANS users. Also, we wanted to show that network-level anonymization is not enough and that there needs to be better education to promote better anonymization.

TABLE II
Vulnerabilities Associated with The CSANS

Vulnerability 1: Intruder able to determine browsing patterns	
Description	An intruder inspects the traffic as it goes through the server and onto the anonymized network
STRIDE Classification	Tampering Information Disclosure

DREAD Rating	Damage Potential = 3
	Reproducibility = 1
	Exploitability = 3
	Affected Users = 3
	Discoverability = 2
	DREAD Rating = 2.4

Vulnerability 2: Intruder attacks the anonymization process	
Description	The intruder attacks the de-anonymization servers and/or the processes associated with the endpoint
STRIDE Classification	Tampering Information Disclosure

DREAD Rating	Damage Potential = 3
	Reproducibility = 1
	Exploitability = 2
	Affected Users = 1
	Discoverability = 1
	DREAD Rating = 1.6

Vulnerability 3: Intruder performs any number of the attacks outlined in the literature review	
Description	By performing any number of the attacks outlined in the literature review
STRIDE Classification	Tampering Information Disclosure Denial-of-Service

DREAD Rating	Damage Potential = 2
	Reproducibility = 2
	Exploitability = 2
	Affected Users = 2
	Discoverability = 2
	DREAD Rating = 2.0

B. Experimental Setup

To accomplish this goal we created documents that were used to both track and infect the CSANS users. We also examined the impact of users performing attributable search queries while on CSANS. We relied on analysis of traffic that we captured to find key points of interest.

The experiment for our case study lasted for two months. There were twenty CSANS users that were included in the experiment and they were asked to conduct all research within the confines of the CSANS.

Prior to the start of the experiment, we created and hosted an inconspicuous website and ensured that both the non-malicious and malicious files were tested before being uploaded as was depicted in Figure 1. Furthermore, we ensured that file logging and tracking was enabled to allow us to capture statistics from the web server. The website utilized various sources for content and was approved by the research manager as being enticing enough to attract the researchers.

The next step was to ensure that the website could actually be found via a simple search. By using Google search engine optimization and similar search ranking tools, we found that our website could in fact be located by virtue of simple search logic. A major aspect of the proposed study design was to determine the attack vector. For the purposes of this experiment, we focused exclusively on Adobe PDF.

The design of our study focuses on three major aspects, namely impact, stealth, and depth. The impact aspect measures how many CSANS users actually accessed the website and clicked on the files. The stealth aspect indicates how many users observed and notified management about the website and/or malicious file. Lastly, the depth aspect quantifies the quality of the malicious file as it relates to the amount and value of PII that a potential intruder could discern.

C. Fictitious Research Task

The CSANS users were specifically asked to complete a research task that focuses on the commercial airline industry. This fictitious research task is like any other task that the CSANS users would typically perform.

To begin the experiment, the research manager provided the CSANS users with a list of keywords to search on as part of the task. To protect the anonymity of the research tasks being done at the case-study agency, the actual and specific keywords have been sanitized and modified to appear more generic. However, to give the reader a frame of reference, we mostly considered keywords associated with the airline industry including: “Airports,” “Flight Times,” “Pricing,” “Mileage Rewards,” “Safety and Security,” and “Jobs.”

The assumption here is that the keywords are general enough to be known by a potential intruder as part of their targeting and profiling. Therefore, simply protecting the keywords is not a viable security measure.

D. Website

The necessary pre-requisite for this experiment was to design and build the aforementioned website. We researched each of the keywords independently and extracted content from other related websites to build into our own custom website.

Since the goal of the experiment is not a demonstration in website design, we developed a basic HTML website and hosted it on a third-party hosting site. The website corresponded to the different topics associated with the keyword list, and using simple logging techniques, the website can log the actions of the users.

The files were uploaded to the “documents” section of the website. The files included documents associated with the keywords. Each file was randomly selected to fall into one of three categories.

The first category was that the files were original and completely clean. The second category was that the files were embedded with a JavaScript (app.launch) function that would automatically open a new webpage in the user’s browser. The file would open a webpage called www.website.com/3490 where 3490 is a random number that can be easily identified in the data logs.

The third category was that the file itself was embedded with malicious code via the Metasploit Framework (see Section IV E). If the malicious code is opened in a vulnerable version of Adobe Reader on a vulnerable Windows OS, the user’s machine could possibly become compromised. Once compromised, it would be trivial for the intruder to de-anonymize the user. Table III shows a sample summary of the document name, the corresponding category, and a description of the exploit where applicable.

TABLE III
SAMPLE SUMMARY OF DOCUMENT AND ACTION

Document (.pdf)	Category	Action
Airports	#1 Clean	Downloads and opens
Flight Times	#2 Redirection	JavaScript app. launch to random page on website
Pricing	#2 Redirection	JavaScript app. launch to random page on website
Mileage Rewards	#3 Malicious	Redirects to webpage being listened to by Metasploit: multi/handler listener
Safety and Security	#3 Malicious	Infected w/ “Aurora” exploit
Jobs	#3 Malicious	Infected w/ “Adobe (Collect_Info)”

E. Metasploit

In order to create the website’s malicious files, we relied on the Metasploit framework [8]. The Metasploit framework was built to provide useful information and tools for penetration testers, security users, and intrusion detection system (IDS) signature developers. The Metasploit framework was created to provide information on exploit techniques and to create a functional knowledge base for exploit developers and security professionals.

The intruder would utilize the Metasploit framework to create attacks that target items such as specific applications, operating systems, and web browsers. The exact steps of finding the exploit, setting and executing the payload, and listening are outside the scope of this paper.

F. Capturing Traffic

We also consider a second threat model associated with the intruder’s ability to capture traffic which is shown in Table IV.

In the ISP case, the anonymized traffic blends with the rest of the regular traffic from subscribers to that particular ISP. Assume that the ISP is Comcast or Cox Communication. There are tens of thousands of users that are accessing the

Internet by virtue of these ISPs. In addition, there are several search queries on any number of topics that are being performed, and this traffic will blend together.

TABLE IV
THREAT MODEL: CAPTURING TRAFFIC

Source of Traffic (Example)	Success / Complexity	Attack Level Success Rate
ISP (Comcast, Cox)	Low / High	Low
Service (Facebook, Gmail, Amazon)	Low / High	Medium
Custom Website	Low / High	High

As it stands, the attack level is low because it would be difficult for an intruder to determine if the search queries and the traffic were from a truly anonymized user or a "regular" user who coincidentally was searching for the same information. Also, it is more difficult to obtain access to a big ISP and monitor all communications. The complexity and cost for doing so is high when compared to poisoning Google or posting an advertisement to CNN.

For example, consider the CSANS user who is researching "Airplane Safety" and finds a website on the "Pan AM Flight 270 disaster" and finds it pertinent to the research task. Now suppose a student who has Cox as their ISP finds the same website for a high school research paper that they are doing. From an intruders point of view, it would be difficult to determine whether either or both the CSANS users and the high school student were attempting to hide their identity given only the traffic obtained from the ISP.

In the external website (service) case, the anonymized traffic blends less when compared to that of the ISP traffic. When someone goes to an external website such as Facebook, Google Mail (Gmail), or Amazon, they give away some of their identity by virtue of logging into a site, accessing personal items, and cookies (e.g. Amazon book preferences). If this traffic is obtained by an intruder, the success rate of determining the identity is increased when compared to simply obtaining the ISP traffic.

For example, consider a CSANS user who accesses their personal Facebook account. The CSANS user must log into Facebook, perhaps views photos, sends a message via Facebook, and posts a few messages. There is something inherently personal about this example. An intruder who is able to acquire the traffic, can often determine the true identity of the user as we will show in the results section of this paper.

In the final case, as was described earlier, we created a custom website that is meant to entice the CSANS users. The website content is associated with a list of keywords that a typical CSANS researcher would use as their search query. Included in this website are a series of files that serve different purposes. The files have the ability to track the users and to also infect the users. The files, if downloaded (and transferred) to their client machine, will potentially give the intruder the ability to discern the user's true identity. It is for this reason that the success rate of the intruder is the highest. We will show the success rate of this novel idea in the results section. The complexity of obtaining traffic from a custom website is also low because there is no dependence on the ISP or popular services.

To substantiate our claims, we also attempted to gather and analyze the traffic that was generated by the researchers while on CSANS. We ran the Wireshark [23] packet analyzer to accomplish this. The generated Wireshark .pcap was then run against a tool called tcpxtract and used to carve headers and footers from the traffic [24]. This extraction yielded several web images, cookies, files, and portions of web activity.

Next, by using a custom-developed Perl script, we were able to take the destination IP address that was captured, and in most cases, identify the hostname, internet registries (e.g. APNIC, ARIN, RIPE), autonomous system (AS) number, the country where the servers are located, the routing information, and the date the website was registered.

We then took all of this information and created a simple Microsoft Access database so that we could perform queries on the data that we found. The database included over 10,000 records as a result of running the experiment and capturing the traffic. The following section describes our findings.

V. EXPERIMENTAL RESULTS

Based on an analysis of the traffic that was generated by the CSANS researchers, we were able to identify a variety of data points that could have been used to identify the CSANS and its users. First, we saw several connections being made from the real (unanonymized) IP address of the CSANS. The source of the IP address was likely a result of the researchers utilizing remote desktop (RDP) from their client (unanonymized) system to access CSANS.

Furthermore, we identified almost daily requests to Facebook. This was a concern because of the obvious attributable nature of this social networking website. We then began to analyze traffic, cookies, and ultimately uncovered two distinct email addresses via the Firefox profiles. Note: It was necessary for us to download SQLlite Database Browser [25] in order to extract this information from Firefox. Going one step further, we logged on to Facebook and did a "Friend Finder" search, based on the two emails that we uncovered. The results of our search yielded the names of two different people who were later shown to have been part of the research group at CSANS.

Next, we discuss the concept of Facebook profile ID. For example, when you login to Facebook (assuming you have not replaced ID with username) you will see something similar to the following whereby the ## correlate to real numbers:

http://www.facebook.com/profile.php?id=604###

During our traffic capture, we were able to identify two distinct Facebook profile ID's. However, when we accessed the profiles, we noticed that it did not relate to any CSANS users. In fact, they were the profiles of two different men living in a foreign country who were not associated with the CSANS organization whatsoever. We still need to perform future research to understand our findings with respect to profile ID. However, it seems to indicate that the CSANS researchers were searching for these two individuals as part of their research task. This has not yet been confirmed or denied. For more information on Facebook ID, the reader is urged to view [26].

The traffic analysis also uncovered the researchers accessing Amazon, Continental Airlines, and several other U.S. based companies. Furthermore, there were other attributable websites such as LinkedIn and Twitter that were being regularly accessed by the CSANS researchers.

By using the tcpxtract tool, we were able to uncover portions of email that were created from Google Mail (Gmail) and Thunderbird. While it was difficult to view the email in its entirety, it appeared to have a subject associated with the Transportation Security Administration (TSA).

By capturing the traffic for such a prolonged amount of time, we were able to find various patterns associated with the time of day users log in to CSANS, the types of sites (both attributable and non-attributable), and the routing of the traffic as it makes its way to these sites. We then compared this traffic to that of non-anonymized traffic acquired from George Mason University (GMU) and found that it was quite easy to tell which traffic patterns were associated with what entity (GMU or CSANS).

In addition to capturing the traffic and utilizing tcpxtract, we also made us of a tool called network miner [27]. This tool allowed us to import a Wireshark .pcap file and mine the traffic. The tool quickly places the traffic into categories such as hosts, frames, files, images, messages, credentials, sessions, DNS, parameters, keywords, clear text, and anomalies. The main results that were acquired from this tool were the search criteria and queries used by the researchers. For example, we found the following search criteria used by the researchers "petrochemical production process," which is an acceptable search query that relates to the given research task. However, we also came across a search query for "Fairfax County School Closings" that is clearly more attributable to the CSANS location.

In the actual fictitious website experiment, we saw the anonymized IP address 207.195.x.x that was identified as Global TAC, LLC. This log entry showed up several times throughout the experiment on any given day. We identified that several of the files from the website were downloaded by this IP address. Shortly thereafter, we could see that our random page was identified in the log files, but this time with the NAT IP address that was confirmed to be that of the target organization. This was repeatable as we saw new IP addresses being related to the CSANS IP address. Also, most of this activity appeared during 8:00 a.m. EST and 5:00 p.m. EST, the timeframe most U.S. based organizations are open for business.

In our logs, we were able to clearly see the web browser and the operating system of the user who opened a certain page in our website. This information could be extended to other de-anonymization projects such as [17]. The intruder could also use this information to customize the Metasploit exploits.

In addition to the logs supplied by our web server, we also utilized tracking statistics from TraceMyIP [19]. This allowed us to have a separate tracking mechanism in place, and we were able to identify on several occasions that a user was in fact logged into our website. We also noticed that both the CSANS anonymized IP address and the case study agency were logged in at the same time. Since the traffic to our website is light, it became quite clear that there was some relationship between the two sets of IP addresses that are simultaneously logged in. This was further supported by the fact that both users logged out of our website at nearly identical times.

Within the logs, we were able to determine whether or not we have seen this person before and in some cases determine the first time they visited and the last time they visited. In addition, we were able to identify the system hardware used and the browser, browser language, operating system, and screen resolution. Next, we could uncover the ISP and NAT IP address of the CSANS organization. Another piece of data that we saw in the logs was the hostname of the CSANS. This hostname appeared to be fictitious as it was the name of a children's game. This information was then looked up via nslookup and came back with an IP address that is tied to the CSANS web proxy. Finally, we identified the connecting city, state, country, time zone, and latitude/longitude.

One particular user was paying close enough attention and identified a suspicious "tracking code icon" that was intentionally displayed on the site. However, as far as we can tell, this was the only time that anyone from the CSANS identified the website or the experiment as suspicious.

Also, we were able to see that on certain days (due to what the IP address was on that given day) that Google Location was able to identify the location of the CSANS organization within about 20 miles. However, on other days, the Google Locator was not able to identify the CSANS location. These results show that at least one CSANS IP address was compromised. This was likely due to attributable search queries being performed. It is not immediately clear whether or not it was a CSANS user who was responsible for the attributable search queries.

Lastly, and most damaging, one user downloaded a malicious file, transferred it to their vulnerable Windows XP computer, and opened it in a vulnerable version of Adobe Reader. By utilizing the Metasploit multi/handler listener, we were able to insert the Metasploit payload and open a shell to the CSANS users system. At this point, we could easily have dropped a keylogger, taken a screen capture of the page, run system commands, and quite easily harvested information that would result in determining the user's true identity.

VI. Discussion and Proposed Defenses

The results of our traffic analysis and experiment were conclusive in determining a few unique CSANS users. In general, the results of the experiment showed that the anonymized IP address could be directly correlated to the CSANS organization. This was shown by looking at the logs and cross referencing a series of random IP addresses with the CSANS organization's NAT address. This information was most likely due to the user downloading the file, transferring it to their computer, opening the file, and allowing JavaScript to run thus causing a random page within the intruder's webpage to launch. In the log files, it was easy to see that a given page of our website was accessed.

Specifically, we were able to conclude that users who transferred the malicious document to their client operating system were at greater risk. This is because the client operating systems and applications were not nearly as secure as the CSANS. In some cases, the client operating system was unpatched and utilizing an obsolete version of Adobe Reader.

Furthermore, we were able to conclude to some degree that attributable searching on CSANS did play a role in determining the location as shown by Google Location. It is not known at this time what the specific search criteria used was, nor is it known whether or not it was a CSANS user who performed the attributable search. We were able to show that given a clean CSANS virtual machine snapshot, certain IP addresses used in the CSANS IP rotation scheme were identifiable as coming from a location in close proximity.

Overall, we were able to show that we can break the CSANS anonymization. Specifically we relied on targeted attacks that focused on the weaknesses of the users and applications. Based on the results and the conclusions, we recommend several possible defenses that the organization, the CSANS, and the users could have implemented to prevent the attacks.

One possible defense would be to reconsider the use of the transfer file system. Even though the transfer file system is a convenient way to move files, it is one of the weakest links in the CSANS security architecture. At a minimum, it is recommended that the users create new and clean PDF files from the ones that they download. If the CSANS organization decided to keep the transfer drive, it is highly recommended that the files be run through multiple anti-virus engines and/or a sandboxed system that could execute the file, trigger on illegal system operations, and therefore potentially block the exploited file from connecting back to the intruder's listener server.

The transfer drive system can be extended with a secure and dedicated analysis platform. An improvement to the system architecture might allow the transfer of files to another virtual machine that would allow for execution but prevent information flows to untrusted hosts. This scheme, similar to what has been done with honeypots, would address the vulnerabilities in the current CSANS, while at the same time, allowing for richer analyses to be performed.

One simpler defense is to patch and harden the client machines. However, we believe that the intruders will continue to create exploits and eventually will be able to infect the client machines. This defense will reduce but not eliminate the risk.

The users should be cautious with allowing JavaScript to run on their computers. Options such as noscript from Firefox [16] will automatically prevent JavaScript from running. The users should have immediately contacted the organization's information security group when their browser started to open new browser windows.

To defend against attributable searching while on the CSANS, the users need to be cognizant that logging into personal accounts such as email, social networking, and banking sites could provide information that could discern their identity. Next, the users are advised not to perform any searches associated with their true geographic location. For example, if the user is located in Denver, CO, they should refrain from searching for items such as Denver sports teams, weather, traffic, jobs, and restaurants. This information can be combined to yield the users identity.

As we demonstrated in section III, Google is already proving this by virtue of their Google service location feature that is tied to a specific IP address regardless of whether that IP address is anonymized [18]. For this reason, we have recommended that the CSANS users complete all of their searches using www.google.com.gh whereby the "GH" is the country code for Ghana. We chose Ghana primarily because it is an English speaking country that is often attributed to the IP anonymization scheme that is used at the CSANS.

Still another possible defense would be to set the Google Location to be a random place in the country. This concept is commonly referred to as "artificial pinning" and is just another small measure that could potentially help to prevent the identity of that particular IP address from being known. At the time of this writing, it was not possible to change the location from outside of the United States assuming that you were using www.google.com (.en) as your main search engine. Also, in some cases, it is not even possible to change this setting and/or it may not appear for a plethora of reasons that are outside the scope of this paper.

Similarly, the users should not provide any personal information to a website, toolbar, or widget about their true location when logged on to CSANS. For example, users often set weather monitoring tools to be their local zip code. The CSANS developers should create visual guidance when a user is using the non-attributable CSANS to search for information related to their true location. Perhaps a flashing notification or a confirmation screen before a given search is executed would be useful.

The overall limitations outlined for privacy-preserving systems are well known, to the point that there are tools such as Privoxy that are designed to supplement the protection offered by ANS. Specifically, Privoxy [33] is a non-caching web proxy with advanced filtering capabilities for enhancing privacy, modifying web page data HTTP, and controlling access. Since it has an application for both stand-alone systems and multi-user networks, one suggestion would be to integrate Privoxy on-top of CSANS.

Another potential solution is to allow regular, non-anonymized users to share the IP address space of the anonymized users. This will create a large quantity of search results that could allow the anonymized users to potentially hide behind an overwhelming amount of search data. For example, if instead of using the CSANS, the organization tunneled a portion of their traffic to universities and colleges in the same area it might actually improve the anonymization. This is because a potential intruder would find it difficult to determine the source of the traffic and whether it was anonymous. Since the research being performed by the CSANS organization might be similar to that of a university professor, it would be nearly impossible to determine which one is anonymous.

VII. Conclusions & Future Work

We conclude this paper by looking at future research. This experiment was done on a small scope with little funding available to the authors. Future research could extend this experiment and consider different anonymization schemes and different applications. We leave it to future research to determine if attacks against other ANS users would be successful. Also, this experiment was narrow with a focus on the commercial airline industry. It would be interesting to determine if a similar attack would work in other organizations such as healthcare or banking. Future research could provide methods to improve the defenses offered in this paper and more importantly help educate the users on how to best defend themselves from the three main attack methodologies.

While not a revelation on its own, we once more show that users and their activities remain the weakest link in any non-trivial security scheme. The implication, however, is the researchers were careless or ignorant of basic security principles. The fact that they betrayed personal information so easily does not reflect well on the level of security education provided to the researchers.

The main takeaway from this study was that both the administrators and CSANS users were not educated about quite prevalent attack vectors for compromising client systems and violating user privacy.

To address this problem, we briefed the CSANS researchers on the results of our findings. We also explained to them the dangers of accessing websites and giving away PII while on the CSANS. The training that we performed was over a two-day span and in it we discussed our findings along with our recommendations as to how to resolve the problem. We are planning on conducting a similar experiment in the upcoming months in order to measure the effectiveness of our training with regards to combating the attacks mentioned in this paper.

Acknowledgment

The authors would like to thank everyone that participated at the case-study agency. The authors would also like to thank Dr. Dwayne Phillips for his editing and suggestions. The authors would also like to thank Javier Almillategui for his help with the development of the custom Perl scripts that were used to capture and sort traffic. Lastly, the authors would like to thank the anonymous reviewers for their insightful comments.

References

[1] Anonymizer [Online]. Available: http://www.anonymizer.com

[2] Tor [Online]. Available: http://www.torproject.org

[3] D. Chaum, “Untraceable electronic mail, return addresses, and digital pseudo-nyms,” *Communications of the ACM*, vol. 4, no. 2, February 1981.

[4] S. Coull, C. Wright, F. Monrose, M. Collins, and M. K.Reiter. “Playing devil’s advocate: Inferring sensitive information from anonymized network traces,” In *Proceedings of the Network and Distributed System Security Symposium (NDSS),* San Diego, CA, 2007.

[5] H. D. Moore and V. Smith, “Tactical Exploitation” [Course Slides], *Black Hat USA 2010*, Las Vegas, NV, July 2010.

[6] N. Schear and D. M. Nicol, “Performance analysis of real traffic carried with encrypted cover flows,” In *Proceedings of the 22nd Workshop on Principles of Advanced and Distributed Simulation (PADS)*, Washington, DC, USA: IEEE Computer Society, 2008, pp. 80–87.

[7] F. Swiderski, and W. Snyder, *Threat Modeling.* Redmond, WA: Microsoft Press, 2004.

[8] Metasploit [Online]. Available: http://www.metasploit.com

[9] X. Wang, S. Chen, and S. Jajodia, “Network flow watermarking attack on low-latency anonymous communication systems,” in *Proceedings of the IEEE Symposium on Security & Privacy (S&P)*, Washington, DC, USA: IEEE Computer Society, May 2007, pp. 116-130.

[10] S. Coull, M.P. Collins, C.V. Wright, F. Monrose, and M. Reiter, “On web browsing privacy in anonymized netflows,” In *Proceedings of the 16th USENIX Security Symposium*, Boston, MA, August 2007, pp. 339-352.

[11] N. Hopper, E. Y. Vasserman, and E. Chan-Tin, “How much anonymity does network latency leak?” in *Proceedings of ACM CCS*, Alexandria, VA, USA, 2007, pp. 13:1 – 13:28.

[12] M. Casado, and M. J. Freedman, “Peering through the shroud: The effect of edge opacity on IP-based client identification. In Proceedings of the 4th USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI ‘07), Cambridge, MA, USA, 2007, pp. 173-186.

[13] S. Coull, C. Wright, A. Keromytis, F. Monrose, and M. Reiter, “Taming the devil: Techniques for evaluating anonymized network data,” In *Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS ‘08)*, San Diego, CA, USA, February 2008.

[14] J. Jung, A. Sheth, B. Greenstein, D. Wetherall, G. Maganis, and T. Kohno, “Privacy Oracle: A System for Finding Application Leaks with Black Box Differential Testing,” In *Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS ’08),* 2008, pp. 279-288.

[15] D. Scott and R. Sharp, “Abstracting application layer web security,” In *Proceedings of the Electrical International World Wide Web Conference (WWW 2004)*, Honolulu, HI, May 2002.

[16] NoScript [Online]. Available:https://addons.mozilla.org/en-US/firefox/addon/722/

[17] Panopticlick [Online]. Available: https://panopticlick.eff.org/

[18] Google Location [Online]. Available: http://www.google.com/support/websearch/bin/answer.py?answer=179386&hl=en

[19] Trace My IP [Online]. Available: http://www.tracemyip.org

[20] Browzar [Online]. Available: http://www.browzar.com

[21] JAP [Online]. Available: http://anon.inf.tu-dresden.de/index_en.html

[22] Safesurf [Online]. Available: http://www.safersurf.com

[23] Wireshark [Online]. Available: http://www.wireshark.org

[24] Tcpxtract [Online]. Available: http://tcpxtract.sourceforge.net

[25] SQLite [Online]. Available: sqlitebrowser.sourceforge.net

[26] Facebook Profile [Online]. Available: http://www.ehow.com/how_5753004_facebook-id.html

[27] Network Miner [Online]. Available: http://networkminer.sourceforge.net

[28] R. Pang, M. Allman, V. Paxson, and J. Lee, "The devil and packet trace anonymization," *ACM Computer Communication Review,*vol. 36, no. 1, pp. 29–38, January 2006.

[29] M. Huber, M. Mulazzani, and E. Weippl, "Tor HTTP usage and information leakage," Springer Berlin Heidelberg: *Communications and Multimedia Security,* vol. 6109, pp. 245-255.

[30] S. Kadloor, X. Gong, N. Kiyavash, T. Tezcan, and N.Borizov, "A low-cost side channel traffic analysis attack in packet networks. In *Proceedings of the 2010 IEEE International Conference on Communications (ICC)*, pp.1-5, May 2010.

[31] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel, "A practical attack to de-anonymize social network users," In *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, Oakland, CA, USA, pp. 223-238, May 2010.

[32] A. Chaabane, P. Manils, and M. Kaafar, "Digging into anonymous traffic: A deep analysis of the tor anonymizing network," In Proceedings 4th International Conference on Network and System Security (NSS), Melbourne, VIC, pp. 167 –174, 2010.

[33] Privoxy [Online]. Available: http://www.privoxy.org

[34] T.G. Abbott, K.J. Lai, M.R. Lieberman, and E.C. Price, "Browser-based attacks on Tor," *Lecture Notes in Computer Science*, vol. 4776/2007, pp. 184-199, 2007.

Framing Effects of Crisis Response Communications on Market Valuation of Breached Firms

Manish Gupta, Raj Sharman, and H. R. Rao

***Abstract*—Research has shown that an adverse security event such as a security breach negatively affects stock price and market valuation of the company. Such events result in substantial loss of reputation and engenders negative corporate image in company's stakeholders. Under such crises, literature has demonstrated that companies devise crisis response strategies to counter the negative effects of the crisis. Use of mass communication has been pivotal in informing the stakeholders of company's post-crisis actions. Media announcements about positive security initiatives are considered as a crisis response mechanism after a security breach. Our research investigates how content of these announcements impact company's stock price. We examine impact of use of certain frames, in media announcements, on stock price.**

***Index Terms*—Security Breaches, Market Valuation, Crisis Response Communication**

I. Introduction

SECURITY breaches and their announcements have significant adverse impact on a firm's market valuation. The breached firms, on average, lose 2.1 percent of their market value within two days of the security breach announcement, which translates to loss of about $1.65 billion per breach in market value of the firm [11]. Privacy Rights ClearingHouse [42] estimates that around 100 million records containing sensitive personal information have been compromised over last few years. A Ponemon study estimates that companies in the UK that suffered data breaches paid £47 per compromised record in 2007 and the average cost per reporting incident was around £1.4m for a company [41].

There are several other studies that have investigated impact of security breaches on stock performance (see, for example, [1][8][23][24][30][31][37]). Companies try to regain some of the losses, both direct such as financial and indirect such as good will, customer confidence, etc. by coming forth with initiatives that will help them rebuild their public image: specifically, announcements related to security improvements and countermeasures can help companies allay some of the adverse impact on their stock performance due to a security breach.

News media coverage has never been more important either in print or electronic form for reputation management [9][10][38]. Announcements about corrective actions re-instill confidence in stakeholders about the company while uplifting and restoring the image that was damaged due to a crisis [43]. There have been several studies reporting reputational capital effect by stock performance of companies [22][25][35]. The crisis response to a security breach incident can take the form of positive announcements regarding company security initiatives to further strengthen overall security posture. After a security breach, positive security announcements such as partnerships with a security services company, a strengthening of authentication for customers [28], or a change in security policies could be the most effective in countering negative publicity from a breach. Companies can assess their posture [26][27] and the causes of a breach then plan initiatives for improvements in security practices.

The research investigates the impact of use of certain frames, in announcements, on stock price. We collected positive security-related announcements in news media announcements made by companies in an attempt to restore image and reputation after a security breach. We examined the impact of use of certain frames in media announcements on stock price. Contributions of the research are two-fold; First, it reveals that companies that experience security incidents should try to make security initiative announcements. Second, it will provide insights into the content of such announcements to maximize the positive impact on stock price. It will guide PR personnel to ensure that announcements about security improvements or investments are made using optimal frames and times to achieve greatest positive impact on market valuations. Given the unprecedented rise in security breaches and their significant negative impact on organizations, the need for corporate crisis response to counter this negativity has never been higher. The rest of the paper is organized as follows: Section II provides background of the area. Section III discusses the methodology of this research in detail, followed by a presentation of the results in Section IV. Section V finalizes the paper with conclusions and directions for future research.

M. Gupta, R. Sharman, and H. R. Rao are with the State University of New York, Buffalo, NY 14260 USA (email: {mgupta3, rsharman, mgmtrao}@buffalo.edu).

II. BACKGROUND

A response is warranted by firms to restore their images and also to mitigate negative outcomes from breaches. This clearly underscores the importance of corporate communications about adverse events and the steps taken or plan to be taken to restore confidence as well as to avoid such events in the future. Companies tend to disclose information that is likely to be positively interpreted by investors with the objective of improving a firm's valuation [50]. Organizations and their stakeholders should attempt to establish "mutual expectations" through their communication" (pp. 166) [47]. Response from companies becomes all the more important in light of the fact that publicity from media is generally considered to be more credible and influential than communications directly from the companies that are planned and may be biased [6]. It has also been found that the negative information about a company is weighted more than positive [39][18], which significantly affects a company's image and how stakeholders interact an organization [4][20] in case of a crisis such as a security breach. An effective corporate crisis response is highly desired for all stakeholders (investors, employees, customers, partners, etc.) to cushion negative outcomes such as loss of legitimacy as well as loss of congruency between company's values (as reflected by its actions) and accepted societal norms [49]. Post-crisis communication has been shown to repair the reputational damage done by a crisis and helps restore corporate image [14]. Stakeholders expect to receive information about a crisis and subsequent response by the organization [22]. Coombs [15] recommend that "crisis managers seek to repair the damage from a crisis, take steps to prevent a repeat of a crisis, or both" (p.180). Benoit [5] explains that bolstering "is used to influence the audience to have a more favorable impression of the source" (p. 80). Crisis response strategies, including what companies say and do post-crisis, have been studied extensively in management literature (See for example [7][44]).

The event-study methodology is a widely-used method in the fields of finance, accounting, and information systems, to study effect of events on market valuations of firms. The premise of event studies is that any new information about a company is efficiently incorporated in the stock price of the company due to changes in perception about expected future performance. Like other event studies, we also use the market model to estimate abnormal returns on the stock prices of companies. The market model assumes a linear relationship between the daily stock returns of a firm and the returns on the market portfolio of its stocks.

Event studies have been used to investigate effects on market valuation of firms due to announcements pertaining to areas such as CIO position changes [13], outsourcing [40][2], IT investments [19][32][33][36], acquisitions of IT companies [48], ecommerce initiatives [45][3], security breaches [1][8][23][24][30][31][37], web portal announcements [29], and information sharing announcements [52], amongst others. Chai et al. [12] looked at information security investments and found that information security investments with commercial exploitation generate more abnormal positive returns than information security investments for IT security improvement. Apart from investigating immediate impact on market valuations, studies have looked into long term negative effects of security breaches [11][46].

III. METHODOLOGY

Having results that indicate that positive security announcements by breached companies help them in image restoration through positive market reaction to such announcements, we investigated if there were specific keywords in announcements that make a difference in terms of magnitude of impact on stock prices. We content analyzed 302 media announcements made by breached companies for keyword frequency. We individually collected impact on stock price due to each announcement using an event study methodology. Eventus® software was used to run 302 outputs with one event for each of announcement. These results from event study were subsequently used for further analysis using content analyses. The results of content analysis was used for logistic regression (using SPSS software) and PLS path modeling (using SmartPLS software). The details and discussions on these are presented in the Results section. We used Atlas.ti ® (version 6.0) for content analysis. We obtained 354,015 words with 24,753 unique keywords. It has been well established in communications studies that frames in communication (words, phrases, images, etc) impact the way the information is presented and received. For instance, to influence readers, the media content feature and emphasize certain aspects of the story [51]. Keywords used for this effect play an important role in the way the message is received, which involves cognitive structures in interpretation [21].

The framing of a message is critical in defining problems, attributions, and solutions [17]. Joslyn [34] asserts that frames (or keywords) help certain facts of values to be projected in a way to make them salient and thus, important for readers. The framing effect is a phenomenon where a communicator emphasizes or uses certain factors to influence opinions and judgments of readers [21]. Coombs [16] has suggested that a crisis manager should emphasize certain cues in post-crisis situations to establish trust and regain reputation. For our analysis, we have used keywords as proxies for these frames – factors and cues that have been shown to help communicators affect the opinions of recipients. We used PLS Path Modeling and Logistic Regression to investigate if the use of certain keywords influence the impact (the stock price - dependent variable). For PLS Path Modeling, we categorized the most frequently occurring keywords into four broad themes: 1) management, 2) technical, 3) security, and 4) stakeholders. Keywords for each theme are as follows: *Management* (business, communication, company, customer(s), enterprise, management, market, sales; *Technical* (applications, mobile, software, technology, wireless); *Security* (access, online,

TABLE I
LOGISTIC REGRESSION PARAMETER ESTIMATES

		B	S.E.	Wald	df	Sig.	Exp(B)	95.0% C.I. for Exp(B)	
								Lower	Upper
Step 1a	Technical_A(1)	0.744	0.427	3.032	1	0.082	2.105	0.911	4.865
	Managerial(1)	-0.603	0.466	1.673	1	0.196	0.547	0.220	1.364
	Interaction(1)	0.265	0.356	0.552	1	0.458	1.303	0.648	2.620
	Constant	0.264	0.749	0.124	1	0.725	1.302		

policy, privacy, secure, security); and *Stakeholders* (industry, provider(s)). *Keywords* were used as "items" or "indicators", while *themes* were "latent variables" and *impact on stock price due to any announcement* was a dependent variable.

IV. RESULTS

The results of the path modeling constituting themes as exogenous as well as endogenous variables are attached as Appendices A and B. Appendix A shows results of the PLS modeling with four latent exogenous variables. Here, we observe that *Technology* and *Management* themes show stronger significance than *Security* and *Stakeholders*. The *Technology* theme has a path coefficient of 3.543 and *Management* has one of 2.789. The keywords in the *Technology* theme include: applications, mobile, software, technology, and wireless. The *Management* theme has the keywords: business, communications, company, customer(s), enterprise, management, market, and sales. In *Technology*, the "mobile" keyword has the strongest loading, while "market" and "business" were strongest for *Management*. Appendix B shows results when *Security* and *Stakeholders* themes were dropped. This resulted in little change for the *Technology* theme, while significance of results for *Management* increased considerably from 2.789 to 3.091. We also investigated only using *Management* and *Technical* themes and results were significant indicating that use of keywords as mentioned in *Management* and *Technical* themes affects the impact on stock price. Additionally, we also analyzed content using non-parametric logistic regression with frequency of managerial and technical keywords as independent variables and impact on stock price as a dependent variable. For specific keywords, frequency higher than mean was coded as 1 and lower than mean as 0. Tables I and II present the results of the analysis that was carried out using SPSS® (version 16) software. The model is of the form of P(Y = y | X = x) where Y is the dependent variable (Impact on stock price) and X are the independent variables (Frequency of *Managerial* and *Technical* keywords).

Table I shows the results of the logistic regression and presents Wald parametric statistical test results of 3.032 and 1.673 for *Technical* and *Managerial* categories. The binomial logistic regression coefficients are 0.744 and -0.603 for these 2 themes. The odds ratios for the predictors (exp(B)) for *Technical* and *Managerial* themes are 2.105 and 0.547, respectively. The p-values of the coefficients for these 2 themes are 0.082 for *Technical* and 0.192 for *Managerial* (Column Sig. of Table I).

TABLE II
PEARSON CORRELATION

Tech	**Pearson Correlation**	1	0.745(**)	0.057
	Sig. (2-tailed)		0	0.385
	N	234	234	234
Mgt	Pearson Correlation	.745(**)	1	-0.046
	Sig. (2-tailed)			0.479
	N	234	234	234
Impact	Pearson Correlation	0.057	-0.0479	1
	Sig. (2-tailed)	0.385	0.479	
	N	234	234	234

Table II shows Pearson correlation coefficients for testing correlation between the two categories. We used 2-tailed tests and results are shown in Table II. A significant correlation is shown between the 2 categories at 0.01 level (2-tailed) with a value of 0.745(**). The significance is at 0.046 with a negative value.

In addition to analyzing the content quantitatively, qualitative approaches were also used. The set of 456 announcements were divided into four groups based on their impact on the stock price of the company that made the announcement. The impact of the top 25% ranged between around 10% and 2%, while the last 25% ranged between -4% to -0.32%. The top keywords in the announcements were grouped in five themes – 1) business, 2) technology, 3) security, 4) management, and 5) communications.

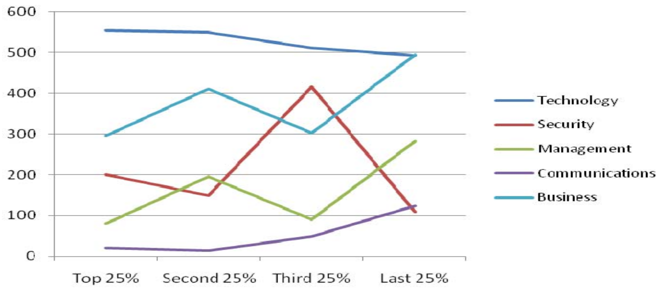


Fig. 1. Qualitative content analysis of announcements: tiered view

Figure 1 shows the top 25% performing announcements used fewer management and business-related keywords, while the last 25% used the most management and business keywords. Also, the top 25% used the fewest communications-related keywords. Collectively, both quantitative and qualitative approaches can cue managers on the keywords that are responded more positively to when security improvement related announcements are made. A different view of the analysis results is shown in Figure 2 that reveals that the last 25% used the most communications, business, and management-related keywords. This implies that investors perceive use of technology and security keywords as trust and image-building endeavors.

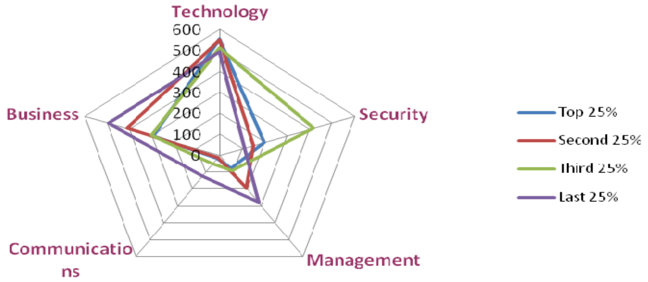
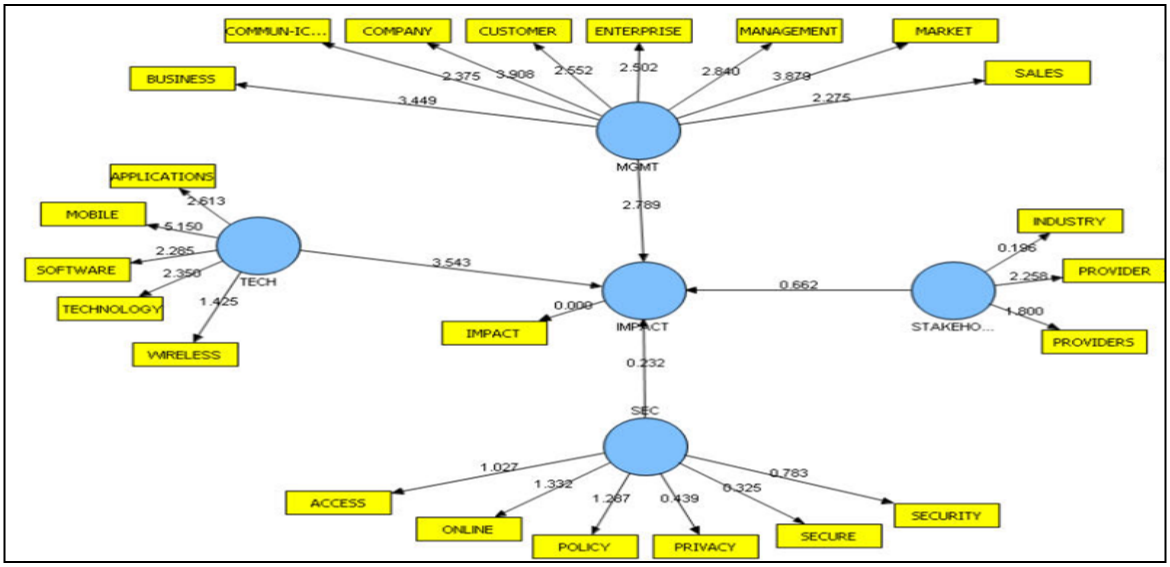


Fig. 2. Qualitative content analysis of announcements – thematic view

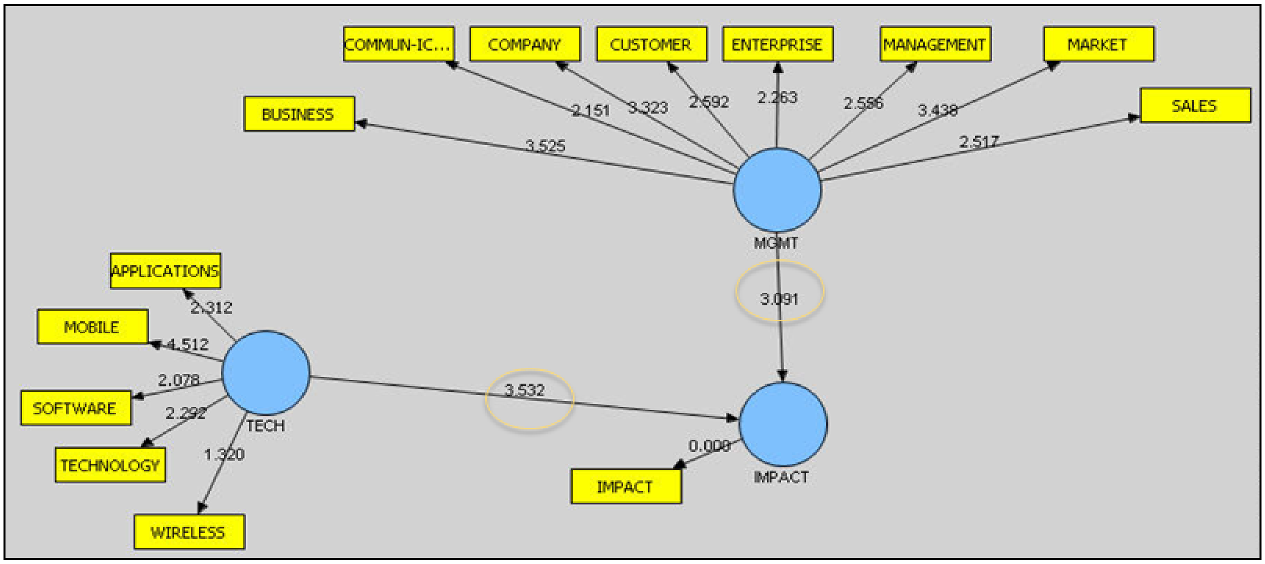
V. Conclusion

The research paper presented analyses around framing effects of announcements made in response to security breaches, by the breached companies, on stock price of the companies. Qualitative analyses presented some strong results and implications of using certain types of keywords in announcements and their influence in changes in stock price. The quantitative studies can be further strengthened to conduct more analyses with different options of keywords in existing and/or emerging categories. In efficient market conditions, the change in stock prices (due to an event) reflect investors' estimation of future profitability and income of the company. This makes this study even more important for managers to understand implications of security breaches and post-crisis response to mitigate any negative impacts. The study is an offshoot of a larger study where we investigated if companies that experienced security breaches (and evidently are adversely impacted by negative stock performance) tended to make security improvement announcements in an effort regain some of the loss, both economic and reputational. Driven by media influence theories, in that larger study, we showed that companies are inclined to make improvements in their security implementations when they are affected by security breaches. Such breaches, for one, show material weaknesses in their security posture that needs to be fixed. In that light, when security initiatives are imperative and almost always necessary, it does make logical sense that they will also, at the same time, attempt to mend the damage done to their market valuation. In that context, we observed that companies are more forthcoming in making efforts in improving their security countermeasures for the benefit of not only shareholders, but also for their customers, employees and partners. The results of that study show that security breaches affect market valuation of companies (based on 260 security breaches from year 2005 to 2009) and that the majority of these affected companies (about 83%) chose to make positive security announcements in an attempt to restore their image and reputation.

Appendix A: PLS Modeling with Four Latent Exogenous Variables



APPENDIX B: PLS MODELING WITH TWO LATENT EXOGENOUS VARIABLES



REFERENCES

[1] A. Acquisti, A. Friedman, and R. Telang, R. “Is there a cost to privacy breaches? An event study,” in *Proceedings of the 5th Workshop on the Economics of Information Security (WEIS)*, University of Cambridge, England, June 2006, pp. 1-23.

[2] N. Aggarwal, Q. Dai, and E. A. Walden, “Do markets prefer open or proprietary standards for XML standardization?” *International Journal of Electronic Commerce*, vol. 11, no. 1, pp. 117-136, 2006.

[3] M. Agrawal, R. Kishore, and H. R. Rao, “Market reactions to e-business outsourcing announcements: An event study,” *Information & Management*, vol. 43, no. 7, pp. 861-873, 2006.

[4] L. Barton, *Crisis in Organizations II, 2nd ed*., Cincinnati, OH: College Divisions South-Western, 2001.

[5] W. L. Benoit, *Accounts, Excuses, and Apologies: A Theory of Image Restoration Strategies*, Albany, NY: State University Press of New York, 1995.

[6] J. Bond, and R. Kirshenbaum, *Under the Radar: Talking to Today’s Cynical Consumer*, New York: John Wiley & Sons, 1998.

[7] J. L. Bradford, and D. E. Garrett, “The effectiveness of corporate communicative responses to accusations of unethical behavior,” *Journal of Business Ethics*, vol. 14, no. 11, pp. 875 – 892, 1995.

[8] K. Campbell, L. A. Gordon, M. P. Loeb, and L. Zhou, “The economic cost of publicly announced information security breaches: empirical evidence from the stock market,” *Journal of Computer Security*, vol. 11, no. 3, pp. 431–448, March 2003.

[9] C. E. Carroll, (2004). “How the mass media influence perceptions of corporate reputation: Exploring agenda- setting effects within business news coverage”, Ph. D. Dissertation, The University of Texas, Austin,Texas, 2004.

[10] C. E. Carroll, and M. McCombs, (2003). “Agenda setting effects of business news on the public’s image and opinions about major corporations,” *Corporate Reputation Review*, vol. 2003, no. 6, pp. 36-46, 2003.

[11] H. Cavusoglu, B. Mishra, and S. Raghunathan, (2004). "The effect of Internet security breach announcements on market value: capital market reactions for breached firms and Internet security developers," *International Journal of Electronic Commerce*, vol. 9, no. 1, 2004, pp. 69-104

[12] S. Chai, M. Kim, and H. R. Rao, (2010). “Firms' information security investment decisions: stock market evidence of investors' behavior,” *Decision Support Systems*, vol. 50, no. 4, March 2011.

[13] D. Chatterjee, C. Pacini, and V. Sambamurthy, “The shareholder-wealth and trading-volume effects of information-technology infrastructure investments,” *Journal of Management Information Systems*, vol. 19, no. 2, pp. 7-42, 2002.

[14] W. T. Coombs, and S. J. Holladay, “Communication and attributions in a crisis: an experimental study of crisis communication,” *Journal of Public Relations Research*, vol. 8, no. 4, pp. 279-295, 1996.

[15] W. T. Coombs, “An analytic framework for crisis situations: Better responses from a better understanding of the situation”, *Journal of Public Relations Research*, vol. 10, pp. 177-191, 1998.

[16] W. T. Coombs, “A theoretical frame for post-crisis communication: Situational crisis communication theory”, in *Attribution Theory in the Organizational Sciences: Theoretical and Empirical Contributions.* M.J. Martinko (ed.), Greenwich, CT: Information Age Publishing, , 2004, pp. 275 – 296

[17] A. H. Cooper, “Media framing and social movement mobilization: German peace protest against INF missiles, the Gulf War, and NATO peace enforcement in Bosnia”, *European Journal of Political Research*, vol. 41, no. 1, pp. 37 – 80, 2002.

[18] E. E. Dennis, and J. C. Merrill, *Media Debates: Issues in Mass Communication*, White Plains,NY: Longman, 1996.

[19] B. L. Dos Santos, K. Peffers, and D. C. Mauer, “The impact of information technology investment announcements on the market value of the firm,” *Information Systems Research*, vol. 4, no. 1, pp. 1–23, 1993.

[20] G. Dowling, *Creating Corporate Reputations: Identity, Image, and Performance*, New York, NY: Oxford University Press, 2002.

[21] J. N. Druckman, “The implications of framing effects for citizen competence,” *Political Behavior*, vol. 23, no. 3, pp. 225 – 256, 2001.

[22] C. J. Fombrun, and C. B. M. van Riel, *Fame & Fortune: How Successful Companies Build Winning Reputations*, New York, NY: Prentice-Hall Financial Times, 2004.

[23] A. Garg, J. Curtis, and H. Halper, (2003a) "The financial impact of IT security breaches: what do investors think?," *Information Systems Security*, vol. 12, no. 1, pp. 22-33, 2003.

[24] S. Goel, and H. A. Shawky, “Estimating the market impact of security breachnext term announcements on firm values,” *Information and Management*, vol. 46, no. 7, pp. 404-410, October 2009.

[25] J. R. Gregory, “Does corporate reputation provide a cushion to companies facing market volatility? Some supportive evidence,” *Corporate Reputation Review,* vol. 1, pp. 288 – 290, 1998.

[26] M. Gupta, S. Banerjee, M. Agrawal, and H. R. Rao, “Security analysis of internet technology components enabling globally distributed workplaces – A framework,” *ACM Transactions on Internet Technology*, vol. 8, no. 4, November 2008.

[27] G. Tanna, M. Gupta, H. R. Rao, and S. Upadhyaya, (2005). “Information Assurance metric development framework for electronic bill presentment and payment systems using transaction and workflow analysis”. *Decision Support Systems,* vol. 41, no. 1, pp. 242-261, 2005.

[28] M. Gupta, H. R. Rao, and S. Upadhyaya. “Electronic banking and information assurance issues: survey and synthesis,” *Journal of Organizational and End User Computing,* vol. 16, no. 3, pp. 1-21, July- September 2004.

[29] M. Gupta, and R. Sharman, “Impact of web portal announcements on market valuations: an event study,” *International Journal of Web Portals,* vol. 2, no. 4, pp. 1-17, 2010.

[30] R. Hasan, and W. Yurcik, (2006). “A statistical analysis of disclosed storage security breaches,” in *Proceedings of 2nd ACM Workshop on Storage Security and Survivability (StorageSS ’06),* Alexandria, VA, 2006, pp.1-8.

[31] A. Hovav, J. and D’Arcy, (2003). “The impact of denial-of-service attack announcements on the market value of firms,” Risk Management and Insurance Review, 6, 2 (2003), 97–121.

[32] S. D. Hunter III, “Information technology, organizational learning, and the market value of the firm,” *The Journal of Information Theory and Application,* vol. 5, no. 1, pp. 1-28, 2003.

[33] K. S. Im, K.E. Dow, and V. Grover, “Research report: a reexamination of it investment and the market value of the firm - an event study methodology,” *Information Systems Research,* vol. 12, no. 1, pp. 103-117, 2001.

[34] M. R. Joslyn, “Framing the Lewinsky affair: third-person judgments by scandal frame,” *Political Psychology*, vol. 24 no. 4, pp. 829 – 844, 2003.

[35] R. F. Knight, and D. J. Pretty, “Corporate catastrophes, stock returns, and trading volume,” *Corporate Reputation Review,* vol. 2, pp. 363 – 381, 1999.

[36] J. Koh, and N. Venkatraman, (1991) “Joint venture formations and stock market reactions: an assessment in the information technology sector”, Academy of Management Journal, Vol. 34 No. 4, pp. 869-92, 1991.

[37] D. Liginlal, I. Sim, and L. Khansa, “How significant is human error as a cause of privacy breaches? An empirical study and a framework for error management,” *Computers & Security,* vol. 28, no. 3-4, pp. 215 – 228, 2009.

[38] M. M. Meijer, (2004). *Does Success Breed Success? Effects of News and Advertising on Corporate Reputation*, Amsterdam: Aksant Academic Publishers, 2004.

[39] R. W. Mizerski, “An attribution explanation of the disproportionate influence of unfavorable information,” *Journal of Consumer Research,* vol. 9, no. 3, pp. 301-310, 1982.

[40] D. Peak, J. Windsor, and J. Conover, “Risks and effects of IS/IT outsourcing: a securities market assessment,” *Journal of Information Technology Cases and Applications*, vol. 4, no. 1, pp. 6–33, 2002.

[41] L. Poneman, “Ponemon evaluates cost of UK breaches,” *Network Security Newsletter,* March 2008, p. 2.

[42] Privacy Rights Clearinghouse. (2009). A chronology of data breaches reported since the ChoicePoint incident (list). *Privacy Rights Clearinghouse* [Online]. Available: http://www.privacyrights.org/ar/ChronDataBreaches.htm.

[43] T. L. Sellnow, R. R. Ulmer, and M. Snider, “The compatibility of corrective action in organizational crisis communication,” *Communication Quarterly*, vol. 46, no. 1, pp. 60-74, 1998.

[44] G. J. Siomkos, and G. Kurzbard, “The hidden crisis in product harm crisis management,” *European Journal of Marketing*, vol. 28, no. 2, pp. 30-41, 1994.

[45] M. Subramani, and E. Walden, “The impact of e-commerce announcements on the market value of firms,” *Information Systems Research,* vol. 12, no. 2, pp. 135-154, 2001.

[46] T. Tsiakis, and G. Stephanides, "The economic approach of information security," *Computers & Security*, vol. 24, no. 2, pp. 105-108, 2005.

[47] G. Turkel, “Situated corporatist legitimacy: the 1980 Chrysler loan guarantee,” *Research in Law, Deviance and Social Control*, vol. 4, pp. 165-189, 1982.

[48] K. Uhlenbruck, M. A. Hitt, and M. Semadeni, “Market value effects of acquisitions involving internet firms: a resource-based analysis,” *Strategic Management Journal,* vol. 27, no. 10, pp. 899–913, 2006.

[49] R. R. Ulmer, and T. L. Sellnow, “Consistent questions of ambiguity in organizational crisis communication: Jack in the Box as a case study,” *Journal of Business Ethics*, vol. 25, no. 2, pp. 143-155, 2000.

[50] R. E. Verrecchia, “Discretionary disclosure,” *Journal of Accounting and Economics*, vol. 5, no. 3, pp. 179-194, 1983.

[51] J. Yioutas, and I. Segvic, “Revisiting the Clinton/Lewinsky scandal: the convergence of agenda setting and framing,”*Journal and Mass Communication Quarterly*, vol. 80, no. 3, pp. 567 – 582, 2003.

[52] Y. I. Song, W. Woo, and H. R. Rao, “Interorganizational information sharing in the airline industry: An analysis of stock market responses to code-sharing agreements,” *Information Systems Frontiers,* vol. 9, no. 2-3, pp. 309–324, 2007.

Automatically Bridging the Semantic Gap using C Interpreter

Hajime Inoue, Frank Adelstein, Matthew Donovan, Stephen Brueckner

***Abstract*—We describe min-c, a C interpreter that solves the generalized problem of the "semantic gap". The semantic gap exists in virtual machine introspection (VMI) and in volatile memory forensics because there is not a native hardware environment. For example, a pointer in a data structure in a process cannot be used without translation to a physical address, a function of the native hardware and operating system. The usual solution is to build an OS interface library to provide the necessary translations. This is brittle as it must constantly track OS versions. Min-c solves this problem by enabling automatic generation of the OS interface library using native OS code itself, or debugging symbols when source is not available. We describe the design of min-c and our method for automatically building the semantic interface database required for type interpretation for both Linux and Windows OSs.**

***Index Terms*—Forensic Memory Analysis, Virtual Machine Introspection, Semantic Gap, Volatile Memory, C Interpreter**

I. Introduction

In volatile memory forensics and virtual machine introspection (VMI) it is necessary to interpret, at a high level, the state of a system which has only been recorded at a very low level. Effective techniques for doing this are increasingly necessary. Most forensics investigations currently are centered on non-volatile storage (hard disks). Hard disk capacities are enormous, however, and the use of encrypted file systems is increasing. Volatile memory can often provide evidence, such as encryption keys, which makes analysis of non-volatile storage faster and easier.

The motivation for VMI is quite different. Sophisticated, stealthy malware (e.g., rootkits) can subvert operating systems entirely, disabling and hiding from the most sophisticated computer security software. It is always possible for the malware to discover, disable, or hide from security software because it runs on the same machine. In light of this, many have proposed and demonstrated approaches based on VMI. This "out-of-the-box" technique allows security software to run in a trusted environment completely outside the OS and the applications it observes.

Both techniques require one machine to interpret the low level state of another machine. VMI applications (we refer to the VM hosting the introspecting application as the host) use the hypervisor to directly access the state of another VM's (the guest) virtual hardware, including the processor, memory, and devices (e.g., disk and network). In volatile memory forensics, the available resource is a core-dump or raw memory image.[1] The difficulty in interpreting this low-level data into a high-level model of the guest system's state is referred to as the semantic gap [1].

The two research communities focusing on this problem have, until recently, been quite distinct. However, we noticed in the course of our VMI research that the problem posed by VMI is identical to that posed in forensics. The goal of our research was to develop a model of the guest's kernel memory space using the semantics of its operating system – the common problem in forensics and VMI. VM introspection libraries supply processor state, and accessing the file system is fairly straightforward [2]. Network operations can also be captured and interpreted easily from outside the VM. The VMI problem is more difficult in that it must not cause side-effects within the guest, and it must also run efficiently, so that real-time monitoring is possible. We therefore concentrate on VMI in this paper, although our technique applies generally to volatile memory forensics as well.

Others (described in Section V) have addressed the semantic gap problem but each proposed solution has limitations. Operating systems may be categorized into major classes (e.g., Windows XP, Vista, Linux 2.4, Linux 2.6), but there is significant variation within each major class that previous efforts do not address. First, they do not account for versioning. Modern operating systems are patched quite frequently, requiring either modification or at the very least, recompilation, of the introspection software. Second, in open source operating systems such as Linux, kernels are customized by distributions or even by individuals themselves.

Due to these two factors, the semantic gap-bridging software

H. Inoue, F. Adelstein, M. Donovan, and S. Brueckner are with ATC-NY, Ithaca, NY 14850 USA (e-mail: {hinoue, fadelstein, matthew, steve} @atc-nycorp.com). This research was supported by the Air Force Research Lab under award FA8750-07-C-0106. The opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the Air Force Research Lab.

[1] For the purposes of this paper, we will refer to the analysis machine as the host and the machine where the memory image was gathered as the guest.

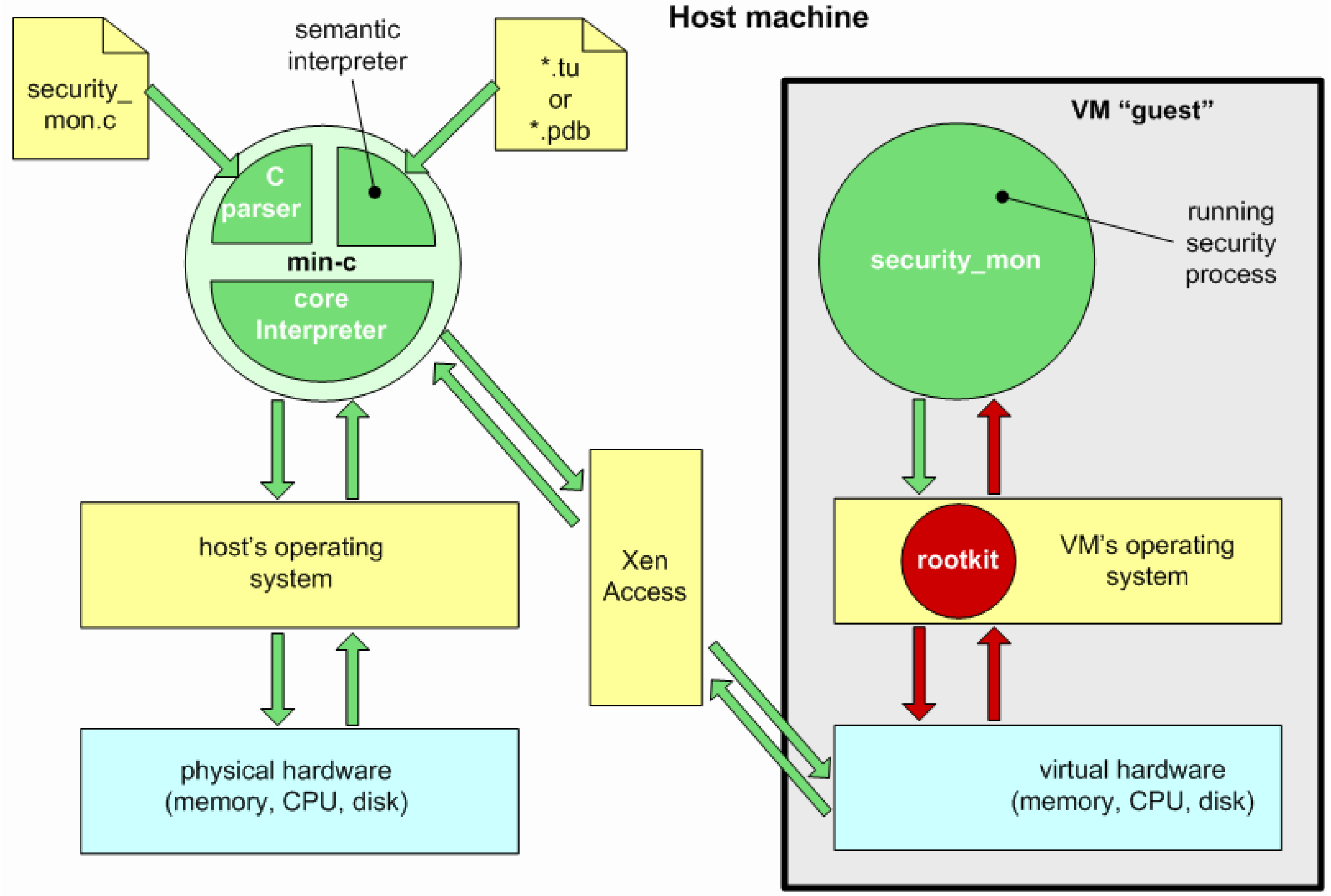


Fig. 1. A comparison of a security monitor using VM introspection with min-c (left) and similar software hosted in the traditional manner (right).

of previous efforts have been prototype implementations written for specific kernel versions that are brittle in the face of changes to those kernels. In addition, each previous effort required manual labor to acquire detailed knowledge of each guest's data types (including the fields of aggregate types) and magic numbers (i.e., memory locations, including relative offsets within aggregate types). This manual labor can include inspection of available files (e.g., symbol tables, header files, source code), reverse engineering, kernel debugging, and trial-and-error coding. For example, Volatility [8], the most popular tool for volatile memory forensics, only supports Windows XP; it does not yet support Windows Vista or Windows 7.

Based on the shortcomings of previous efforts, we determined that a useful solution for bridging the semantic gap should be both general and automatic. That is, it should enable us to run any version or distribution of a major OS class without recompilation of the introspection code and without the need for manual intervention in the process.

Finally, a solution that bridges the semantic gap should minimize the distinction between guest and host. It should be easy to reuse and port security software. Developers should not need to learn an introspection API or language, as previous solutions have required. In short, we want our introspection layer to be invisible—host code should look and run as if it were in the guest.

Given these desiderata, we implemented a C interpreter, linked with an introspection library. It currently runs a large subset of the C90 standard.[2]

In addition, we have generated semantic reconstruction libraries that allow us to automatically locate and properly interpret data structures in both the Linux and Windows kernels. We call our semantic gap-bridging software min-c, which integrates our C interpreter and semantic reconstruction libraries. Fig. 1 shows how we use introspection and min-c to achieve our objectives and compares them to traditional non-introspective security software running inside a guest.

Min-c is an abbreviation for "EXAMIN-C." EXAMIN is a commercial project with the goal of developing a testing platform for containing, triggering, analyzing, and reverse engineering stealthy malware. It is a VM-based workbench based on the Xen hypervisor that employs VM introspection to provide high-assurance detection of stealthy malware on both Windows and Linux platforms. EXAMIN incorporates a number of practical tools based on our introspection

[2] C90 was chosen for its ease of implementation. We may switch to C99 in the future. The interpreter does not currently support some data types, such as ones we have not encountered in the Linux and Windows kernels (e.g., floating point).

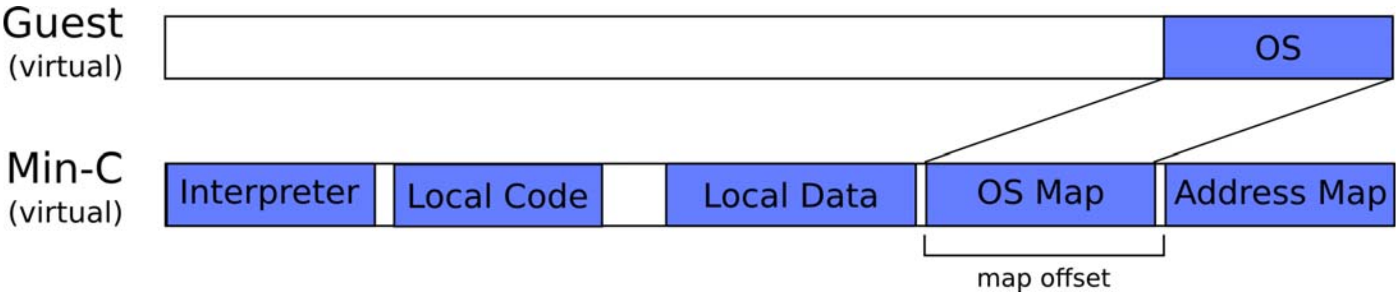


Fig. 2. Memory layout of a min-c application. Interpreter translates pointers by subtracting map offset stored in address map from pointers in mapped regions.

techniques, including integrity monitors and cross-view checkers.

Min-C forms the core of EXAMIN. In the rest of the paper, we describe how min-c solves the various problems involved in practical VM introspection and bridging the semantic gap. We begin by presenting our min-c design. We first describe how we automate the gathering of kernel-specific semantic information and then how we integrate this information with our C interpreter. Next, we describe our current applications of min-c. Finally we discuss the limitations of our approach and describe future and related work before concluding the paper.

II. Design

The min-c interpreter consists of the following four components, each of which we subsequently describe in detail:

1. the introspection library,
2. the C parser,
3. the core interpreter, and
4. the semantic interpreters.

We used the open source XenAccess library to provide VM introspection of the Xen hypervisor [3]. We modified XenAccess to provide the ability to memory map a range of contiguous virtual memory from the guest into the host's address space. This is the principal introspection mechanism. XenAccess also provides access to files, enabling volatile memory forensics investigations as well.

We built the C parser with the aid of the TreeTop parsing library for the Ruby programming language [4]. The C parser reads in the source text and generates custom bytecode. Our parser performs type checking, but our custom bytecode instruction set does not embed type information such as size or offset information. Instead, type information is retrieved (for host datatypes) or calculated (for guest datatypes) during execution, because the version of the OS is not determined until runtime.

The core interpreter executes the bytecode generated by the parser. It uses the XenAccess library to map guest kernel memory into the host's address space and then operates on that memory similar to the way it operates on unmapped memory. The only exception is in the use of pointers: the address space mapped into the host makes pointers inconsistent. A variable referred to by a pointer in the guest has a different address on the host. The min-c interpreter keeps track of which portions of memory are mapped in from the guest and translates pointer addresses appropriately.

Fig. 2 shows how the map of the physical memory of guest OS is mapped into the process space of min-c. In the figure, the OS memory is contiguous in physical memory, which is usually the case. It is also possible to map the virtual address space of a user process, which is usually not physically contiguous, into a min-c application. A min-c application has five

logical memory areas: the mapped memory from the guest, the min-c interpreter itself, the min-c application's code (local code), its heap (local data), and the address map.

The address map is used by the interpreter to translate pointer addresses. It is a table which describes the mapping between the virtual address space of the guest and the virtual address space of the min-c application. When a min-c application reads a pointer, the interpreter consults the table to determine if it is in a guest-mapped space. If it is, then the pointer is read from the mapped space and then the map offset between the two spaces is added so that the new pointer is accurate. Note that the interpreter must understand that the value being read is a pointer. Applications that treat pointers as integers will not have these values properly translated. Thus, developers need to take particular care with types or memory corruption will immediately result.

Our semantic interpreters provide type and layout information for data structures within the guest. Semantic view reconstruction begins with an address and a type. If we have information about the type, we can then understand the region of memory specified by the address in terms of operating system semantics. If the type contains a pointer, we can then recursively apply this procedure to the address and type referred to by the pointer. In this way, we can rebuild the semantic meaning of the address space within the guest.

In Section V, we describe the manual and debugger-oriented procedures previous efforts have used to generate a semantic view of the guest's memory. As discussed in Section I, the procedures used in previous efforts are brittle and difficult to use with heterogeneous systems. In order for our semantic interpreters to automatically reconstruct a semantic view of the observed guest kernel, we need the data structure layout created by the compiler. For Linux, compiling the observed kernel with debug symbols would give us access to this information, but unfortunately this would change the resulting image. For Windows, source code is not available so recompilation is also not an option. We extract the data structure information using a unique procedure for each major class of operating system, as follows.

For Linux, the GCC compiler can output a kernel's intermediate representation as a flat file database (.tu) in text format using the --fdump-translation-unit flag [5]. Since source code and kernel configurations are available for each Linux distribution, we can recompile each kernel of interest in this manner a single time, and use the resulting .tu files for all subsequent introspection activities. We do not need to replace the original guest kernel with the recompiled one because they are identical (We delete the recompiled one, saving only the .tu files generated by the compilation process). The objective is to generate the equivalent of debugging information without having to compile the kernel with a debugging flag. The .tu files give us access to the type, size, and layout information needed to automate semantic view reconstruction. Addresses are not available in the intermediate representation (these are generated by the linker), so we use the System.map file generated by the kernel compilation process (and usually exported in the /boot directory) to obtain them. The System.map file gives us exported addresses, but does not give us other important addresses, such as the location of the system call table, which we obtain through forensic processes that are automatable for each major kernel version.

For Windows, the compiler generates a Program DataBase file (.pdb) that contains the necessary symbol and type information [6]. An executable file (.exe or .dll) stores the name of its associated .pdb file as well as the version (specified by a globally unique identifier and age value). PDB files can be made available by an application developer, or in the case of Microsoft's executables (such as the Windows XP, Vista, and 7 kernels), are downloadable from Microsoft's Windows Symbol Server. Dynamic Link Libraries (DLLs) export public symbols for use by other applications but non-public symbols are stripped out during the compilation process. PDB files contain both public and private symbol information, such as addresses, as well as function signatures and their locations. Addresses are stored as Relative Virtual Addresses (RVAs) that are simply offsets from the start of the file when loaded into memory [7]. PDB files also contain data type information similar to that available in Linux .tu files.

Not all relevant information is provided by PDB files. Many internal data structures are not described, nor are many addresses which are useful for VMI or forensics. When structures or addresses are not available, developers in min-c can supplement this with files written in C. This is a key advantage of min-c, since it allows developers to augment the interface library and analysis scripts in the same language they use to write native OS code itself, and in a way that makes calls to the interface library implicit, as we show in the next Section.

Fig. 1 shows how four parts (the introspection library, C-parser, core interpreter, and semantic interpreter) compose the min-c interpreter and conduct semantics-aware VM introspection of a guest's kernel. A complete program is created when min-c extracts, interprets, and combines the source code (.c) file with the semantic information database (.tu or .pdb) files. The core interpreter uses local resources to execute, but redirects introspection queries to the guest via XenAccess. It accomplishes this by tracking and maintaining consistency of separate data types and pointers for each domain.

TABLE I
PROGRAMS FOR LISTING LOADED MODULES
WITHIN A GUEST VM RUNNING LINUX 2.6 KERNEL

XenAccess Specific Partial Listing

```
xa_read_long_sym(&xai, “module”, &next_module;
list_head = next_module;

while(1) {
   memory=xa_access_virtual_address(&xai, next_module, &offset);
   if(memory == NULL) {
      perror(“failed to map memory for module list pointer”);
      goto error_exit;
   }
   memcpy(&next_module, memory+offset, 4);

   if(list_head == next_module)
      break;

   /* Note – the module struct that we are looking at has a string directly
following the next/prev pointers.  This is why you can just add 8 to get the
name.  See include/linux/module.h for more details. */
   name = (char *)(memory+offset+8);
   printf(“%s\n”, name);

   munmap(memory, xai, page_size);
}
```

Min-c Equivalent

```
/* Pull address from System.map */

extern struct list_head module* modules;
struct list_head* next_module = modules;

while(1) {
   struct module tmp;
   next_module = next_module->next;
   if(modules == next_module)
      break;

   tmp = list_entry(next_module, struct module, list);
   puts(tmp->name);
   puts(“\n”);
}
```

On the top is a partial source listing that uses only the XenAccess library. On the bottom is the mini-c version. Note: We ignore locking in this example.

III. Current Applications

We currently use min-c within our EXAMIN platform for cross-view checking and integrity checking. Rootkits hide their presence from both user-space and kernel-space applications. Cross-view checking tools compare the state of the guest as reported from within (the guest) with that reported from without (the host) using VM introspection. Differences often indicate that stealthy malware has infected the host.

The min-c interpreter makes it much easier to write cross-view checking tools. To illustrate, consider Table I, which shows the listing for VM introspection code that outputs the list of modules running in a Linux system. Using only the XenAccess VM introspection library, the code is awkward, and relies on hardcoded offsets for linked-list pointers and the name string. Min-c, however, automatically reconstructs a guest kernel's datatypes and offsets, so its code is identical to that written for a kernel module. It is shorter, easier to understand, easier to write, and will run on many different versions of Linux, since the structure is interpreted at runtime, instead of compiled.

Our integrity checking tools monitor portions of the guest kernel's memory to detect tampering. During execution, the kernel code and many of its variables should not change frequently after initialization. For our EXAMIN implementation, integrity checking tools repeatedly poll code and selected structures, such as the system call table or interrupt descriptor table, for modification. Changes can indicate a rootkit infection.

Our integrity checking tools can also take advantage of min-c's semantic reconstruction capabilities. Rather than simply detecting a change to a data structure, we can describe the change. For example, when the system call table is “hooked,” min-c can provide the name of the specific call that was altered and locate the memory space to which it now points.

IV. Discussion

The min-c interpreter effectively achieves our goals; it automatically bridges the semantic gap in a manner that is general to major kernel classes. Users do not have to learn a new API or language, and can use the native language (C) of the operating system to write scripts that look like kernel code running in the guest.

We have rewritten the larger examples provided with XenAccess in min-c style.[3] We have scripts identical to code written for the kernel, which execute properly for multiple versions of Windows and Linux. Our scripts list current processes and drivers (modules in Linux). We have developed cross-checkers for the system call tables for Windows and Linux which will inform administrators when the system call table has been modified, and which system calls have been hooked, which can help identify which rootkit is responsible.

This functionality forms the basis of our EXAMIN system, which is intended as a malware incubator. We can easily write new monitoring scripts for the system. Debugging is quite easy; we find that we can write Windows drivers and Linux modules to monitor kernel state directly, and then run them in min-c.

EXAMIN is clearly beneficial as a security and reverse engineering tool. It also can be useful as a tool to aid digital forensic analysis, in particular analysis involving live systems and volatile data, such as live memory. The interest in live memory analysis is growing rapidly, and while many advances have been made in recent years, there are relatively few tools to bridge the semantic gap.

Volatility [8] is probably the best known tool to conduct memory analysis, but until recently it relied on a pre-existing memory image. Simply getting the memory image can be challenging [9]. We became aware during the implementation of min-c that the Georgia Tech group responsible for XenAccess added hooks that enable Volatility to access live VMs, giving it similar introspective capabilities to min-c [10]. Similarly, XenAccess also supports access to memory dump files. While Volatility currently has more analysis tools than min-c, we believe that the min-c approach is superior because new tools for Volatility must be written in Python and use explicit translation libraries. Min-c can make writing new tools much easier. New system analysis tools will use or reuse code that is almost identical to the equivalent kernel C code. It also allows a single tool to target multiple versions of an OS, because the interpreter links in the appropriate translation library at runtime. With Volatility, this is not possible.

Because EXAMIN uses VM introspection, it has essentially no impact on the running system and is very unobtrusive. These qualities are highly desirable for forensic analysis [9]. EXAMIN can help analysts conduct an investigation of a running system to find data that may only reside in memory, or may locate data that are essential for a traditional disk analysis, such as whole-disk encryption keys, whose absence render the data on a disk useless.

A. Limitations

There are still several limitations to min-c that hinder its application to other problems.

First, we cannot read guest memory that is paged out to disk. This is not a problem with our current EXAMIN objective of kernel monitoring because kernel memory for the kernel structures that are required by our tools is never paged out.[4] To monitor guest memory for user-space applications would require modification of the guest: injecting code to induce page faults, causing the desired application pages to be read back into core. Although this is possible, our current goals are to perform introspection without directly tampering with the guest, allowing security services to remain invisible.

Currently, no volatile forensics applications support examination of paged-out memory, either. It may be more straightforward to implement support for this for forensics, where everything is on a file system, than on an executing system where memory and the file system monitoring must maintain consistency.

Also, because our introspection method is based on polling, we cannot detect changes when they are quickly reversed. This possibility becomes less likely if our scripts poll more

[3] The minor examples supplied by Xen Access are implicitly provided by min-c functionality.

[4] Volatility does not have this ability, either.

often, but the cost is decreased performance. We can fix this problem using memory access alerts (Section IV-B), but this will require modifications to the Xen hypervisor.

Because persistent malware is much more likely to be detected through file system virus checkers or similar software, we believe memory-only rootkits are becoming more common, and the min-c approach is most effective against them.

Third, the performance of min-c is limited by the fact it is interpreted rather than compiled. There are two reasons for choosing to run interpreted. First, data structure layout differs depending on patch-level in Windows and kernel configuration and compiler version in Linux. It is convenient to have one introspection application script that is useful for every guest of a major class instead of requiring new compiled versions for every kernel upgrade. This could be partially mitigated by having a just-in-time (JIT) compilation system, but this still would not mitigate the second problem: that of pointer translation. We map guest kernel memory into the min-c address space on the host using Linux's mmap. Our interpreter properly converts pointer values from addresses on the guest to addresses in the min-c address space. Because the introspection library does not allow us to specify a target address for mmap, we cannot calculate this address ahead of time. We could fix this with more complicated logic in a JIT compiled approach if greater performance is required.

Fourth, min-c is strictly a C interpreter. It does not interpret C++ code, nor can it import C++ datatypes (classes) for either Linux or Windows. This is acceptable for our current applications because the target is the kernel, which is typically written in C.

B. Future Work

Our min-c interpreter is still incomplete. There are still several features we believe will make it more useful.

Two features would enhance min-c for both digital forensics use and VMI:

JIT Compilation

We plan to port min-c to the LLVM framework [11]. LLVM is a toolkit for writing interpreters, virtual machines, and compilers. LLVM would vastly increase performance and allow us to support many languages.

OS Fingerprinting

At the moment, the operator must specify the correct version and patch level for the guest in order for the interpreter to identify the appropriate .pdb or .tu files needed to interface with it. We intend to automate this so that version information is automatically deduced by min-c on startup. On Linux, this information is usually available in the /proc/version file and is represented in memory by the init_uts_ns variable. Unfortunately, this variable's address differs by version and configuration, so it is not straightforward locate it and perform the check. The process is easier on Windows because the executable files themselves store the name and version of the associated .pdb file. It is a simple manner to parse the executable on disk to correctly identify the correct .pdb file. Pagel has described an effective method for fingerprinting Windows [12]. This is particularly important for digital forensics. While it is likely that operators will know what operating systems are running in their VMs, forensics investigators often receive images with no other information describing them.

Three other potential features would be VMI specific, and would require modifications to the hypervisor or XenAccess layer:

Synchronized Access

Access to many OS data structures is synchronized using locks. If a structure is locked (being modified) when we attempt to read it, it may be in an inconsistent state, causing our interpreter to make incorrect semantic interpretations. An appealing approach is to lock the data structures from the interpreter, allowing us to properly access these extended data structures. Our interim method is to pause the guest and check that the data structure is unlocked. If so, we read it, otherwise we briefly execute the VM and try again. Note that this procedure can act as a spin lock.

Memory Access Alerts

A highly desirable feature would be the ability to raise an alert when the guest writes to, reads from, or executes within a specified address range. This alert could cause the execution of an arbitrary script. It would also obviate the need for polling and enable EXAMIN's integrity tools to immediately discover when changes are occurring and prevent rather than detect intrusions. Memory alerts on read or execute operations could act as breakpoints, allowing min-c to operate as a scriptable debugger. VMware's VMSafe introspection library supports memory triggers, but requires that a VM be booted in a special introspection mode [13]. It is impossible to start or stop monitoring during VM execution. We prefer XenAccess's ability to start and stop monitoring of any VM at any time.

Replay

Finally, we consider the ability to rollback execution of the guest to any point in the past to allow examiners, when anomalies are discovered, to hunt down their origins. This would be similar to the ReVirt system [14], but with usability perhaps more similar to UndoDB [15].

V. RELATED WORK

Most research on the semantic gap is specific to VMI. The Volatility project appears to be the basis of digital forensics research on volatile memory research. There has been recent work towards automated generation of this OS library. Case, Marziale, and Richard [16] demonstrated automatic generation of the Volatility OS library using .pdb files. Okolica and Peterson [17] use a similar strategy. Our work is more general, in that we also enable specification using C, which allows us to support Linux and other operating systems. The C preprocessor in particular allows us to support multiple OSs in a compact, clear way.

The VMI techniques described in this paper require access to a guest VM's state via the hypervisor. The open source project XenAccess [3] facilitates this process for the Xen hypervisor, and VMware's VMsafe [18] provides access to information from some of VMware's hypervisors. Both projects are fairly young and do little more than acquire the state of virtual hardware and do little to bridge the semantic gap with the guest OS, although XenAccess includes some sample modules that interpret structures in kernel memory.

The problem of creating high-level semantics from low-level hardware information acquired by VM introspection was identified by Chen and Noble [1], who applied the term “semantic gap.” An early solution for bridging the semantic gap was implemented by Garfinkel and Rosenblum [19] in their Livewire prototype. The approach used a Linux crash dump analysis tool [20] as an “OS interface library.” However, this approach applies only to kernels which have been compiled using non-standard flags (including debug symbols). Our approach allows us to run the same kernels that ship with standard Linux distributions, without requiring recompilation. Also, their tools are specific to Linux while our approach works with Windows kernels as well.

Many techniques useful in VM introspection are common to digital forensics. Several toolkits have been written that interpret physical memory images. Toolkits include idetect [21], Windows Memory Forensic Toolkit [22], the Volatility Framework [8], VADTree [23] and FATKit [24]. Others try to search memory directly, without fully rebuilding semantic representations. This is called “memory carving”; Schuster's DFRWS '06 paper is an example [25]. Carving is typically focused on a limited set of non-kernel datatypes and does not deal with memory layout, and is therefore insensitive to specific OS version and patch level. However, it is much less capable than the previous approaches that try to bridge the semantic gap by interpreting kernel structures, and therefore only support a limited range of OSs.

A common technique to bridge the semantic gap is to locate known structures in memory (by symbol table lookup, access to source code, or by scanning memory for matches) and then traverse and interpret these structures. This technique is used by [26], [27], [28], [29], [30], and [31]. These efforts rely on the manual process of locating ``magic numbers'' (structures addresses, their internal data types, and relative offsets), and writing the equivalent of kernel code to traverse and interpret them. Furthermore, this manual process must be repeated for different kernels and kernel versions, which frequently change due to new releases, patches, re-compilation, etc. This challenge is acknowledged by Hoglund [32] and Jiang et al. [2].

EXAMIN's feature set is similar to a research system built by Jiang, Wang, and Xu [2]. Their paper also details the difficulties in bridging the semantic gap, and describes a solution known as “guest view casting.” In their description they state, “Configuration variation over the same OS... adds additional complexity to VM semantic view reconstruction. However, the guest view casting methodology remains effective despite these differences, as shown by our evaluation...” This statement is technically true, but does not reflect the considerable difficulties in building a general approach that systematically monitors a range of guest types. The methodology is effective, but tedious, time-consuming, and requires that it be redone for every new version.

Most VM introspection research to date uses semantic reconstruction of the guest's state to acquire information (e.g., module or process lists) or to integrity check static memory structures. A more sophisticated use is to detect unauthorized tampering with dynamic memory structures, but this is a more challenging problem. Petroni's group tackles this problem by implementing a high-level language for the specification of “security predicates” [33]. The language allows them to specify constraints or invariants that indicate a security fault if violated. Our work is similar in that it allows us to monitor the guest with a high-level language. However, we believe that using C's flexibility allows developers to easily specify high-level invariants as well as work at the lower level of direct memory access.

In min-c, we have used a strategy that allows developers to create OS interface libraries automatically where symbols or source are available, or in the easiest manual fashion through specification in C. Recently machine learning (ML) approaches have begun to emerge. Payne [34] and Dolan-Gavitt [35] showed how classifiers could be trained to recognize structures from multiple versions of Windows. Kolbitsch et al. [36] have begun extracting algorithms from raw binaries. We see min-c as complementary to ML. Where source or symbols are available, automatic generation of the OS interface is always preferable because of the error rate of ML algorithms. Where it is not, algorithms that output C data structures will allow developers to easily analyze, correct, and store their output.

VI. Conclusion

We presented min-c, a C interpreter that bridges the semantic gap problem facing applications of virtual machine introspection and volatile memory forensics, and showed its applications to VMI-specific security monitors. Like other approaches to VMI, min-c provides tamper resistance by moving security software out of a monitored VM and into the host system. Unlike other approaches, existing security monitor source code requires minimal changes because min-c makes the VM introspection invisible to the code, interpreting data structures and pointers appropriately during execution. Our approach is general within major operating system classes and automated to reduce the need for manual reverse engineering, making it attractive to both forensics and VMI applications. We believe this approach has great promise in furthering the migration of security software from monitored VMs to less vulnerable host systems and as a platform for volatile memory forensics.

REFERENCES

[1] P. M. Chen, and B. D. Noble, "When virtual is better than real," in *Proceedings of the Eighth Workshop on Hot Topics in Operating Systems (HOTOS '01),* Washington, DC, USA: IEEE Computer Society, 2001, p. 133.

[2] X. Jiang, X. Wang, and D. Xu, "Stealthy malware detection through vmm-based 'out-of-the-box' semantic view reconstruction," in *CCS '07: Proceedings of the 14th ACM Conference on Computer and Communications Security*, Alexandria, VA, October 2007, pp. 128– 138.

[3] B. D. Payne, M. Carbone, and W. Lee, "Secure and flexible monitoring of virtual machines," in *Proceedings of the 23rd Annual Computer Security Applications Conference (ACSAC 2007)*, Miami Beach, FL, December 2007, pp. 385-397.

[4] N. Sobo. Treetop [Online]. Available: http://treetop.rubyforge.org/index.html.

[5] R. M. Stallman, and the GCC Developer Community, *Using the GNU Compiler Collection*. Boston, MA, USA: GNU Press, 2003.

[6] Microsoft Corporation. Visual studio pdb files [Online]. Available: http://msdn.microsoft.com/ en-us/library/yd4f8bd1(VS.71).aspx.

[7] S. B. Schreiber, *Undocumented Windows 2000 Secrets*. Upper Saddle River, NJ, USA: Addison-Wesley, 2001.

[8] Volatile Systems. Volatility framework [Online]. Available: https://www.volatilesystems.com/default/volatility.

[9] H. Inoue, F. Adelstein, and R. A. Joyce, "Visualization in testing a volatile memory forensics tool," To be published at the *2011 Digital Forensics Research Workshop*, Aug 2011.

[10] B. Dolan-Gavitt, B. Payne, and W. Lee, "Leveraging forensic tools for virtual machine introspection," Georgia Institute of Technology, Atlanta, GA, USA, SCS Tech. Rep. GT-CS-11-05, 2011, pp. 1-6.

[11] C. Lattner, "Llvm: An infrastructure for multi-stage optimization," Master's thesis, Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA, December 2002.

[12] B. Pagel, "Automated virtual machine introspection for host-based intrusion detection," Master's thesis, Engineering and Management, Air Force Institute of Technology, Wright-Patterson AFB, OH, USA, March 2009.

[13] VMware. Vmsafe partner program overview [Online]. Available: http://www.vmware.com/technical-resources/security/vmsafe/ security_technology.html.

[14] G. W. Dunlap, S. T. King, S. Cinar, M. A. Basrai, and P. M. Chen, "Revirt: enabling intrusion analysis through virtual-machine logging and replay," in *Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI '02)*, vol. 36, no. SI, pp. 211–224, 2002.

[15] Undo Software Ltd. Undodb - reversible debugging for linux [Online]. Available: http://www.undo-software.com/

[16] A. Case, L. Marziale, and G. G. Richard III, "Dynamic recreation of kernel data structures for live forensics," in *DFRWS '10: The Proceedings of the 10th Annual Digital Forensic Research Workshop (DFRWS'10)*, Portland, OR, USA, August 2010, pp. 41–47.

[17] J. Okolica, and G. L. Peterson, "Windows operating systems agnostic memory analysis," in *DFRWS '10: The Proceedings of the 10th Annual Digital Forensic Research Workshop (DFRWS'10)*, Portland, OR, USA, August 2010.

[18] VMware. VMsafe security technology [Online]. Available: http://www.vmware.com/ overview/security/vmsafe.html.

[19] T. Garfinkel, and M. Rosenblum, "A virtual machine introspection based architecture for intrusion detection," in *Proceedings of the Network and Distributed Systems Security Symposium (NDSS '03)*, 2003.

[20] Mission Critical. Linux. Crash core analysis suite utility [Online]. Available: http://oss.missioncriticallinux.com/projects/crash/.

[21] M. Burdach. (2005, July 11). Digital forensics of the physical memory [Online]. Available: http://forensic.seccure.net/pdf/mburdach_digital_ forensics_of_physical_memory.pdf

[22] M. Burdach. (2005, July 9). An introduction to windows memory forensics [Online]. Available: http://forensic.seccure.net/pdf/ introduction_to_windows_memory_forensic.pdf

[23] B. Dolan-Gavitt, "The vad tree: A process-eye view of physical memory," in *Proceedings of the 8th Digital Forensic Research Workshop (DFRWS '07)*, Pittsburg, PA, USA, August 2007, pp. 62–64.

[24] N. Petroni, A. Walters, T. Fraser, and W. Arbaugh, "Fatkit: A framework for the extraction and analysis of digital forensic data from volatile system memory," *Digital Investigation, the International Journal of Digital Forensics and Incident Response*, vol. 3, no. 4, pp. 197-210, December 2006.

[25] A. Schuster, "Searching for processes and threads in microsoft memory dumps," in *Proceedings of the 7th Digital Forensic Research Workshop (DFRWS '06)*, Lafayette, IN, USA, August 2006, pp. 10–16.

[26] F. D. Baiardi, and S. Sgandurra, "Building trustworthy intrusion detection through vm introspection," in *Proceedings of the 3rd IEEE International Symposium on Information Assurance and Security (IAS '07)*, Manchester, UK, August 2007, pp. 209-214.

[27] M. Bergdal, and T. A. Sorby, "Using virtual machines for integrity checking," Master's thesis, Informatics, University of Oslo, Oslo, Norway, 2007.

[28] X. Jiang, and X. Wang, "'Out-of-the-box' monitoring of VM-based high-interaction honeypots," in *Proceedings of the 10th International Symposium on Recent Advances in Intrusion Detection(RAID '07)*, Queensland, Australia, September 2007, pp. 198-219.

[29] R. Jones. Virt-mem: Tools for monitoring virtual machines [Online]. Available: http://et.redhat.com/~rjones/virt-mem.

[30] N. Petroni, T. Fraser, and W. Arbaugh, "Copilot—a coprocessor-based kernel runtime integrity checker," in *Proceedings of the 13th USENIX Security Symposium (SSYM'04)*, San Diego, CA, USA, Aug 2004, pp. 179-194.

[31] R. Riley, X. Jiang, and D. Xu, "Guest-transparent prevention of kernel rootkits with vmm-based memory shadowing," in *Proceedings of the 11th International Symposium on Recent Advances in Intrusion Detection (RAID '08)*, Cambridge, MA, USA, pp. 1-20.

[32] G. Hoglund, "DARPA: Rootkit detection," HBGary, Tech. Rep., 2007.

[33] N. Petroni, T. Fraser, A. Walters, and W. Arbaugh, "An architecture for specification-based detection of semantic integrity violations in kernel dynamic data," in *Proceedings of the 15th USENIX Security Symposium*, Vancouver, B.C., Canada, July/August 2006, pp. 289-304.

[34] B. D. Payne, "Improving host-based computer security using secure active monitoring and memory analysis," Ph.D. dissertation, Computer Science, Georgia Institute of Technology, Atlanta, GA, USA, 2010.

[35] B. Dolan-Gavitt, A. Srivastava, P. Traynor, and J. Giffin, "Robust signatures for kernel data structures," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, Chicago, IL, USA, November 2009, pp. 566-577.

[36] C. Kolbitsch, T. Holz, C. Kruegel, and E. Kirda, "Inspector gadget: Automated extraction of proprietary gadgets from malware binaries," in *Proceedings of the 31st IEEE Symposium on Security and Privacy*, Oakland, CA, USA, May 2010, pp. 29-44.

Protection Profile-Based Scenario-Centric Taxonomy of Secure Routing Protocols in Ad hoc Networks

Mohammad Iftekhar Husain and Ramalingam Sridhar

***Abstract*—Secure routing bears significant importance in wireless ad hoc networks where nodes act both as terminals as well as routers. Although a large body of research exists in literature to address this issue, the variations among the approaches are overwhelming. This necessitates a well-defined taxonomy of existing routing protocols. Current taxonomic approaches for secure routing in ad hoc networks typically do not account for varying security goals and environmental characteristics of different forms of application scenario. In this paper, we present a scenario-centric taxonomy of secure routing protocols in ad hoc networks. We devise a novel framework to define a protection profile for application scenarios that incorporates the notion of security goals, possible attacks and environmental characteristics simultaneously. Finally, we classify the existing secure routing protocols according to their suitability in application scenarios with varying protection profiles. Comparison with existing taxonomic approach shows the effectiveness of our method in terms of completeness, usability and extendibility.**

***Index Terms*—Ad hoc networks, Routing protocols, Security, Classification.**

I. Introduction

An ad hoc network consists of a set of nodes that carry out basic networking functions like packet forwarding and routing without the help of an existing infrastructure or centralized administration. Nodes in an ad hoc network rely on one another for forwarding a packet to its destination, primarily due to the limited range of each host's wireless transmissions. Such interdependency of communication among nodes makes routing in ad hoc networks more vulnerable compared to wired networks.

Several protocols have been proposed in the literature to secure the routing operation in ad hoc networks. Some of the protocols are proposed to address particular attacks. For example, Watchdog and Pathrater [1] are proposed to address routing misbehavior. Some of them just act as a security extension to existing routing protocols such as Secure Ad hoc On-Demand Distance Vector routing [2]. There are also variations in techniques used to achieve security goals as well. For example, message integrity can be ensured by hash or message digest. These factors give an impression that the problem space is vast and hard to explore and address. So, a necessity to classify secure routing protocols arises from the following reasons:

- To choose a protocol appropriate for an application scenario,
- To understand the similarities and differences among protocols,
- To be able to assess the feasibility, effectiveness and cost of deployment, and
- To compare different protocols to each other.

Existing secure routing protocols for ad hoc networks can be classified based on different symmetric, asymmetric and hybrid cryptographic security mechanisms employed to provide security services such as authentication and user integrity. Most often such efforts [3], [4] are based on security mechanisms tailored for specific routing protocols and cannot be used interchangeably with other routing protocols. Secure routing protocols are also classified based on a range of security threats in ad hoc networks. For instance, security protocols such as CONFIDANT [5] are specifically designed to address routing attacks and misbehaviors. However, none of the existing solutions focus on classifying the secure routing protocols according to the intended application scenario of an ad hoc network. The rationale behind such classification criterion is the fact that different forms of ad hoc networks have different security goals and environmental factors.

In this paper, we propose a protection profile based scenario-centric taxonomy of secure routing protocols in ad hoc networks inspired by the notions of the *Common Criteria* [6], the federal criteria for information technology security evaluation proposed by National Security Agency (NSA). We present a framework to define the protection profile for different forms of ad hoc networks. Based on the protection profile, we then analyze the suitability of existing routing protocols for these scenarios. We also compare our taxonomic approach with existing ones in terms of completeness, usability and extendibility.

II. Secure Routing Protocols

In this section, we briefly discuss some representative secure routing protocols proposed in the domain of ad hoc networks.

Authenticated Routing for Ad hoc Networks (ARAN) [7] is a stand-alone secure routing protocol for ad hoc networks. It uses cryptographic certificates to provide authentication and non-repudiation. So, it requires the existence of a trusted Certificate Authority (CA). Each node, before attempting to

M. I. Husain is a PhD student of Computer Science and Engineering, University at Buffalo, The State University of New York, Buffalo, NY 14260, USA (e-mail: imhusain@buffalo.edu).

R. Sridhar is an Associate Professor of Computer Science and Engineering, University at Buffalo, The State University of New York, Buffalo, NY 14260, USA (phone: 716-645-3186; fax: 716-645-3464; e-mail: rsridhar@buffalo.edu).

connect to the ad hoc network, must contact the CA and request a certificate for its address and public key. The protocol assumes that the public key of the CA is pre-distributed among the nodes. Routing traffic messages, such as route discoveries and route replies, must be signed by the node that generates or forwards them. It follows an on-demand approach for basic routing operations.

Secure Efficient Ad hoc Distance vector routing (SEAD) [8] follows a Destination-Sequenced Distance-Vector (DSDV) [9] approach for routing. In order to find the shortest path between two nodes, the distance vector routing protocols utilize a distributed version of the Bellman-Ford algorithm. It uses hash chains to authenticate hop counts and sequence numbers. Generally, a hash chain is created by applying a one-way hash function to a random value repeatedly. SEAD uses the elements of this hash chain to secure the routing traffic. This protocol requires the existence of an authentication and key distribution scheme for security operations.

Ariadne [10] is based on Dynamic Source Routing (DSR) [11] and came from the same researchers who proposed the SEAD protocol. It assumes the existence of a shared secret key between two nodes. Ariadne also uses a message authentication code (MAC) in order to authenticate point-to-point messages between these nodes. TESLA broadcast authentication method [12] is used to authenticate broadcast messages related to routing. The usage of TESLA mandates the presence of strict time synchronization as well.

Secure Link State routing Protocol (SLSP) [13] is a protocol that can be deployed as stand-alone solution for proactive link-state routing, or combined with a reactive ad hoc routing protocol to be used in a hybrid framework. SLSP uses an asymmetric key pair for every a node by which it secures the discovery and the distribution of link state information. Participating nodes are identified by the IP addresses of their interfaces. However, key management and routing misbehavior are not addressed by the authors.

Secure Ad hoc On-Demand Distance Vector routing (SAODV) [2] is a combination of security extensions to the standard Ad hoc On-Demand Distance Vector (AODV) [14] protocol. It utilizes cryptographic signatures to authenticate the non-mutable fields of the messages. The route discovery process is secured using a one-way hash chain. SAODV also assumes the existence of a key management scheme.

Secure Position Aided Ad hoc Routing (SPAAR) [15] uses geographic information and asymmetric cryptography to provide routing security. Geographic information is also used to make forwarding decisions, which reduces routing traffic significantly. Although computation intensive, SPAAR meets most of the security requirements of a scenario.

The CONFIDANT [5] system consists of a set of extensions to DSR that is comprised of a monitor, a reputation system, a path manager and a trust manager. Routing paths are chosen based on the reputation of the nodes which is calculated by monitoring the behavior of that node.

The Watchdog and Pathrater [1] scheme consists of two extensions to the DSR routing protocol. Watchdog is responsible for monitoring the next node in the path whether it forwards the packet properly. Nodes who fail to do so are identified as misbehaving node by watchdog. The pathrater assesses the results of the watchdog and selects the most reliable path for packet delivery. However, insider attack is not considered in this approach.

Packet Leashes [16] are security extensions that can be used with an existing routing protocol to protect against wormhole attacks. It requires strict time synchronization, or a combination of loose time synchronization and the knowledge of geographical location through GPS.

A. Limitations of Current Secure Routing Protocols

Although a number of secure routing protocols have been proposed in the current literature, these solutions have severe limitations when applied to ad hoc network scenarios with varying topologies. Most of these protocols are developed to target a specific threat or a pre-defined network domain. In ad hoc networks, due to the network dynamics, there is a need to develop and adapt routing protocols according to the demands of the risk and threat level and available resources. Below, we detail some of the factors that limit the applicability of current secure routing solutions to ad hoc networks.

Attack-based protocols: Most of the existing secure routing protocols are aimed at providing solutions to specific types of attacks. For instance, the SAR protocol solves eavesdropping and routing table attacks, but not for other attacks on routing protocols. On the other hand, Ariadne, does not offer any protection against eavesdropping attack. Current solutions are not often equipped to protect against all possible routing threats. This is particularly challenging since ad hoc networks are susceptible to constantly changing attacks and with intelligent adversaries introducing novel attacks.

Key distribution mechanism: Some secure routing protocols such as SAR and SPAAR are based on cryptographic security primitives. However, most of these protocols do not outline or discuss methods for key distribution. As key distribution and management is challenging in ad hoc networks without the presence of any centralized control, we cannot realistically assume the availability of cryptographic keys as proposed in these solutions.

Network environments: Security protocols are mostly developed specific to a network environment. Several assumptions are often made regarding centralized control, trusted entities and available network resources with no clear perspective of the deployment scenario. For instance, the ARAN protocol requires the existence of a trusted CA to authenticate ad hoc routing traffic. Since current protocols do not account for varying characteristics of ad hoc network applications, they do not provide a stand-alone security solution for all target network environments.

Lack of standard security evaluation: A key problem in most of the existing security protocols is lack of standard evaluation criteria. Since several solutions are based on different evaluation criterion, it is not feasible to compare and analyze their security performance. Also, lack of standard platform for performance evaluation further detriments the study. Different simulators used with different simulation metrics do not provide accurate results.

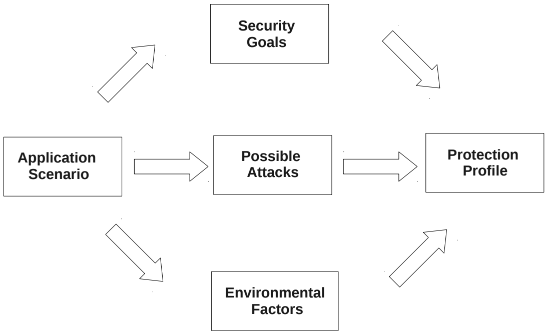


Fig. 1. Components comprising a protection profile. An application scenario embodies the security goals, possible attacks and environmental factors. These three compose the protection profile.

III. Protection Profile of Network Scenarios

Current secure routing protocols use a range of security mechanisms to address different security threats in ad hoc networks. However, as discussed in the previous section, existing key management and secure routing protocols do not provide solutions against all possible security threats. Existing methods are often unrealistic as they are independent of the network and environmental assumptions and hence lack the capability to act as stand-alone secure protocols in these networks. In order to ensure complete security solutions, it is necessary to design a framework that can clearly define a consistent set of policies and procedures needed to protect the network systems.

Common Criteria Model: The structural formation of our framework is inspired by the notions of the *Common Criteria* [6], the federal criteria for information technology security evaluation. The Common Criteria introduced the notions of protection profiles and security targets. According to the framework, a user would generate a protection profile to detail the protection needs, both functional and assurance, for a specific situation or a generic scenario. The protection profile would be abstract specification of the security aspects needed in an information technology product. The idea of defining ad hoc security scenarios based on a realistic security framework is similar to that of a protection profile. In response to a protection profile, the vendor would then map the requirements of the protection profile in the context of the specific product onto a statement called a security target. The security target then becomes the basis for the evaluation.

IV. Components of Protection Profiles

Our classification framework comprises the protection profile that specifies the security requirements and objectives for varying ad hoc network scenarios. The framework components are shown in Figure 1. For any given application scenario, one can determine the security goals, possible attacks and environmental characteristics of that particular ad hoc network. Based on these components, one can identify a well-balanced protection profile for the ad hoc network application scenario in consideration. Below, we describe each of the components in our classification framework.

A. Application scenario

Due to their minimal configuration and rapid deployment, ad hoc networks are suitable for use in various applications ranging from battlefield combat scenarios to sensors collecting patients data in a hospital. The prime component of our framework is application scenario as it forms the basis for developing a security solution. Security requirements, network characteristics and possible security threats vary depending on the intended use of ad hoc network applications. Hence, security solutions must be developed to adapt to the needs of different ad hoc network based applications.

B. Security Goals in Ad hoc Networks

As with traditional networks, to secure ad hoc networks, it is essential to consider the following attributes: confidentiality, integrity, authentication, availability and non-repudiation. These attributes define the security goals in a network.

Confidentiality (CONF): Confidentiality guarantees that the transmitted information is not disclosed to unauthorized entities. It ensures that the routing data from the source node must be accessible only to the intended destination. Leakage of sensitive information to adversaries may have disastrous consequences. Hence, care must be taken such that even if the information is tapped by an adversary, the information is not easily decipherable, thus assuring confidentiality. The most common security mechanism for achieving confidentiality is encryption. Encryption transforms a message into a ciphertext using an encryption key. Some of the most popular cryptographic encryption methods are Public Key Infrastructure (PKI) and symmetric key encryption. In wireless networks, nodes use cryptographic keys to encode and decode confidential messages.

Integrity (INTG): Information integrity ensures that the transmitted message is not altered or corrupted. When a message is transmitted in wireless medium, the data can be modified, tampered or deleted by malicious nodes in the network. The most significant security requirement for the routing mechanism concerns integrity. In case of military applications, routing information may be tactical information of primary importance. Thus, while considering security for routing control messages, we mostly consider how to generate and verify digests or digital signatures. The most popular techniques used to achieve integrity include Checksum, Cyclic Redundancy Check (CRC), Message Authentication Code (MAC) and Message Integrity Code (MIC). In wireless channels, MAC/MIC are more commonly used to verify data integrity.

Authentication (AUTH): Authentication provides the ability to identify a node and prevent node impersonation. Without authentication, a node in a wireless network can easily masquerade as another node gaining unauthorized access to the network resources. Common methods used for authentication are username-password, shared secret verification and biometrics.

Availability (AVAL): Availability ensures the survivability of the network under the presence of Denial-of-Service (DoS) attacks. It is important for a network to remain operational at all times to keep the network services and resources available to legitimate users. DoS and routing attacks targeting network availability can be mitigated by use of hash chains. For instance, hash chains can be used for numbering the packets. By limiting the number of packets by hash or cryptographic chains, DoS attacks such as flooding can be prevented. Intrusion detection mechanisms can also be used to detect attacks affecting availability. For example, the Watchdog mechanism can be used to promiscuously listen to the wireless medium and detect next hop node misbehavior. This method is used to detect packet drops in the network. Similarly, the Pathrater detection mechanism uses an average of the nodes' rating to evaluate the quality of the path to detect misbehavior. Path quality rating is compiled from link breaks, active nodes (where a packet was successfully sent in a previous time interval) and watchdog accusations. In general, it is possible to detect and isolate misbehaving and selfish nodes in an ad hoc network through intrusion detection techniques. Particular attacks that may be detected using these solutions are Black Hole [17], Wormhole [18] and Byzantine [19] attacks.

Non-repudiation (NREP): Non-repudiation guarantees that the sender of a message cannot later deny having sent the message. Digital signatures, a mechanism using PKI can be used as a process to provide NREP. However, the calculation overload should be taken in consideration when deployed in ad hoc networks.

C. Attacks on Ad hoc Networks

In this section, we present some of the attacks that target the above mentioned security goals in ad hoc networks.

Attacks targeting confidentiality (ACONF): In this attack, a malicious node can intercept and receive conversations of legitimate nodes. Since the wireless communication medium is broadcast in nature, transmitted messages can be easily overheard and fake messages injected into the network.

Authentication attacks (AAUTH): Impersonation or spoofing is a type of attack where the adversary assumes the identity of an authorized node. By impersonating another node, the adversary can receive data transmitted to the nodes and gain access to network resources that may not be available to them under normal circumstances.

DoS attacks (AAVAL): This type of attack is launched to deny network access to authorized entities. A simple DoS attack can be launched by flooding packets to targeted node(s) in the network. In wireless networks, DoS attacks can be launched at multiple protocol layers. In physical and MAC layers, an adversary could employ jamming signals to disrupt the on-going transmissions on the wireless channel. In the network layer, an adversary can exploit the routing protocol to disrupt the normal functioning of the network. In higher protocol layers, a malicious node could bring down critical services such as the key management service.

Environmental attacks (AENVR): In hostile environments, a malicious node can launch various routing attacks to disrupt routing operations or deny service to the network. Below, we list some of the routing attacks in ad hoc networks affecting network availability: (a) *Routing Table overflow*: In this attack targeted on table driven routing protocols, a malicious node advertises routes to non-existent nodes in the network thereby filling up routing tables and resources of legitimate nodes in the network. An adversary performs routing table poisoning by sending fictitious routing updates or modifying genuine route update packets sent to other uncompromised nodes. In packet replication, the malicious node replicates stale packets consuming additional bandwidth and battery power resources. (b) *Wormhole attack*: In this attack, a pair of colluding attackers receives and records packets at one location and replays them at another location in the network. Due to the broadcast nature of the radio channel, the attacker can create a wormhole even for packets not addressed to itself. (c) *Black hole attack*: In this attack, an adversary falsely advertises shortest path or stable path to the destination node during the route discovery process. The intention of this attack is to disrupt the path-finding process or to intercept all data packets being sent to the destination node concerned. (d) *Byzantine attack*: A group of intermediate malicious nodes works in collusion and creates routing loops, routing packets to non-optimal paths, or dropping packets. This attack is difficult to detect as the network seems normally operational to the other nodes.

D. Environmental Factors

Different applications of ad hoc networks have varying network characteristics. For instance, nodes in these network may differ in terms of mobility, resource constraints and bandwidth requirements. Secure routing protocols used in such networks should take these fundamental differences into consideration. Hence, in our scenario centric framework, environmental assumptions acts as a critical component necessary for modeling and developing secure routing protocols according to the demands of the target network. The following environmental factors are considered:

- GPS Capability: Yes (GCAP), No (GDIS)
- Backbone Infrastructure: Yes (BBP), No (BBA)
- Node Capability: High (NCH), Low (NCL)
- Energy Constraints: High (ELH), Low (ELL)

Table I summarizes these environmental factors for different network scenarios.

TABLE I
ENVIRONMENTAL FACTORS FOR DIFFERENT SCENARIOS

Scenario	GPS Capability	Backbone Infrastructure	Node Capability	Energy Constraints
TANET	Yes	Yes	High	Low
VANET	Yes	Yes	High	Low
AANET	Yes	No	High	Low
WSANET	No	No	Low	High
UWANET	No	No	Low	High
CDNET	No	No	Low	High
BANET	No	No	Low	High
FRANET	Yes	Yes	Low	High

E. Protection Profile

A protection profile is used to define and specify security requirements for a given system to address the problems without dictating how these requirements will be implemented. It is often a combination of threats, security objectives, assumptions, security functional requirements, security assurance requirements and rationales. Since ad hoc networks are deployed in different applications and varying network scenarios, protection profiles can be used to define security requirements and protection needed to secure these networks. It is possible that a security breach in any layer can facilitate a potential attack on the entire wireless network. Secure routing protocols in ad hoc networks should be based on the protection profile to specify the type of protection suitable for the network in consideration. Specifically, the protection profile for secure routing in ad hoc networks comprises the following items:

- Assumptions about security aspects of the environment in which ad hoc network is deployed,
- Security threats to be addressed by the network,
- Security goals and objectives for the network, and
- Rationale demonstrating how the requirements meet the security goals, and how the security goals address the threats.

V. Protection Profile Examples

A protection profile comprises a range of security mechanisms that can used to meet the security objectives of different wireless ad hoc networks. Below, we describe some of these according to the scenario.

Tactical Ad hoc Network (TANET) is a network formed for communication among soldiers in a tactical operation. It is not feasible to set up a fixed or wired network for communication in a hostile environment such as the battlefield. TANET provides a required communication mechanism in this kind of environment. Due to the sensitive nature of communication involved in the tactical battlefield, these networks require the highest level of security and reliability. The TANET environment is comprised of heterogeneous nodes such as low-capability sensors, medium-capability nodes for data delivery, or high-performance nodes with directional antennas that relay network traffic via satellite links or airborne backbone network. Hence, in most cases, connection with a central server is possible for certification, authentication and security credential update. Also, since nodes in these networks are often equipped with GPS, it enables accurate localization and position estimation.

All five aspects of security goals are required for the nodes in TANET. Due to the nature of tactical operations, confidentiality of any message exchanged is of topmost priority. Only authenticated nodes should be able to communicate with each other and integrity of the message should also be ensured. Nodes should maintain their availability for communication and measures should be taken so that nodes cannot repudiate a transaction. Location privacy should also be maintained.

Vehicular Ad hoc Network (VANET) provides communication among vehicles to exchange information such as traffic, weather, road condition and accidents. VANET allows vehicles to avoid problems either by taking cautionary actions or alerting the driver.

Nodes in VANET are characterized by high mobility and driver behavior. These networks do not have a fixed infrastructure and hence rely on vehicles for network functionality. Vehicular nodes communicate with road side controllers which in turn exchange information with some central authority. Due to sufficient power supply from car batteries, energy efficiency is not an issue in VANETs.

The confidentiality requirement in VANET is flexible compared to TANET. However, only legitimate nodes should be allowed to communicate with each other and messages should be exchanged intact ensuring integrity.

Airborne Ad hoc Network (AANET). In this kind of network, network devices such as routers and transceivers are carried on high altitude aircrafts and uninhabited aerial vehicles (UAV). These networks sometimes act as the backbone for terrestrial ad hoc networks such as TANET.

Airborne network nodes are mobile with high velocity. They often use ground stations or satellite links for communication. GPS based localization and position estimation is available in this network as well. Nodes in this network maintain constant communication with a trusted central authority.

As this kind of network mostly works as a backbone for other ground ad hoc networks, availability is the topmost priority. Only authenticated nodes should be allowed to participate in the network. Integrity of the message should also be ensured.

Wireless Sensor Ad hoc Network (WSANET). Wireless sensor motes collecting data in different environments such as volcanic eruption and firefighting form this kind of ad hoc network. Data collected by the motes are sent to a central node for further processing.

Nodes in this network are inherently resource constrained with low processing speed, storage capacity and limited communication bandwidth. Once deployed, sensor nodes remain static in most cases.

The main task of nodes participating in this network is data aggregation. So, ensuring integrity of messages is of paramount importance. Also, non-repudiation mechanisms should be forced to find out if any node is sending bogus values.

Underwater Ad hoc Network (UWANET) networks are formed for localized monitoring and coordinated networking amongst a large amount of underwater nodes. Currently, UWANET is widely used for uninhabited ocean monitoring.

Underwater networks differ from generic ad hoc and sensor networks in terms of huge propagation delay, low bandwidth and use of acoustic signals for communication. Sometimes, they also use autonomous underwater vehicles for better communication. Nodes in these networks often have low or medium mobility due to environmental water current. Nodes in this network can depend on surface stations for security operations. However, due to resource constraints, node level processing of cryptographic operations may not be feasible.

Nodes in this network are also data collecting sensors. Maintaining integrity is very important in this network as well. Only authenticated nodes should participate in the network.

Collaborative and Distributed Ad hoc Network (CDANET) fulfills the requirement of applications that need temporary communication with minimal configuration among a group of people in a class or conference. For example, researchers in a conference can use this network to exchange presentations. An instructor in a classroom can use CDANET for distributing lecture materials.

Nodes in this network are often laptops and personal hand held devices. However, due to the sporadic nature of this network formation, assuming the presence of any central authority might not be a feasible option.

As this network is formed to share data among nodes of a certain group, there should be mechanisms in place to block outsider nodes from participating in network activities. Integrity of the exchanged data should also be ensured.

Body Area Ad hoc Network (BANET) consists of mobile sensors implanted on the body that communicate with each other to monitor vital body parameters. BANET is mostly used at hospitals to monitor critical patients.

Small medical sensors are often used in this kind of ad hoc network. These nodes can possess shared secrets before deployment, but due to resource constrained sensor devices, we cannot perform expensive cryptographic computation on these nodes.

In most cases, data in this network is very sensitive and highly private medical information. So, mechanisms ensuring both confidentiality and integrity should be in place. Also, authentication should be done before any network operation is executed.

First Responder Ad hoc Network (FRANET). During disaster or accidental emergency scenario, this kind of network helps first responders to communicate with each other and exchange information. Other emergency scenarios include search and rescue, crowd-control and fire-fighting.

Nodes in this network are heterogeneous. Some of these are hand held devices used by the first responders. These hand held devices communicate with the sensors measuring the condition of the injured. Devices held by the responders can communicate with a central authority, but sensors do not have that capability.

The emergent characteristic of this network requires solid mechanisms to ensure availability. Also, confidentiality should be maintained as the data is the vital information of injured persons in most cases.

VI. Protection Profile based taxonomy of Secure Routing Protocols

In this section, we classify the secure routing protocols according to the protection profiles of different scenarios. According to our investigation, most of the protocols are not a perfect fit for the protection profile of a given scenario. However, we have found that the combination or slight modification of existing protocols makes them a close fit for the scenarios.

Tactical Ad hoc Network (TANET): The SPAAR protocol provides confidentiality, integrity, authentication and non-repudiation. The environmental assumption of this scenario such as the presence of GPS is also suitable for the operation of this protocol. However, to address routing misbehavior, either CONFIDANT or Watchdog-Pathrater should be used. Packet leashes should be used to address wormhole attacks.

Vehicular Ad hoc Network (VANET): The environmental assumption and protection profile of VANET makes ARAN and SAODV suitable for the dynamically changing nature of the network. Both of them are on-demand routing protocols. However, availability related attacks are not addressed in these routing protocols. On the other hand, the SPAAR protocol can also be used in this network if the vehicle is GPS enabled.

Airborne Ad hoc Network (AANET): A SLSP routing protocol providing authentication, integrity and non-repudiation is suitable for this scenario. As this network works mostly as a backbone network, the network topology does not change frequently. So, a link state routing protocol meeting the requirements of the protection profile is a good fit for AANET. SLSP also provides protection from availability related attacks such as DoS attack, which is critical for a backbone network.

Wireless Sensor Ad hoc Network (WSANET): Although multiple secure routing protocols meet the security goals of WSANET protection profile, environmental characteristics of this network makes them an infeasible choice for it. However, by using a lightweight shared secret pre-deployed among participating nodes, SEAD- and ARIADNE-like protocols might be a close fit for this scenario. However, protocols such as ARAN and SLSP that need access to a central authority for their function are not suitable for this scenario.

Underwater Ad hoc Network (UWANET): Similar to WSANET, with the existence of a pre-deployed shared secret, SEAD- and ARIADNE-like protocols might be used in this type of network as well. When autonomous underwater vehicles are present, central authority based protocols might be used.

Collaborative and Distributed Ad hoc Network (CDANET): SAODV closely fits the protection profile of the scenario. In addition, a simple routing protocol using MAC or MIC along with a shared secret works for the CDANET protection profile.

Body Area Ad hoc Network (BANET): Although, SPAAR meets the security goals of BANET's protection profile, the environmental assumption of this network makes SPPAR an infeasible solution for this network. Shared secret based lightweight symmetric encryption in conjunction with MAC/MIC can be used in a routing protocol to fit the protection profile of this network.

First Responder Ad hoc Network (FRANET): The Protection profile of this network makes SPAAR a good choice for this scenario if the hand held devices of first responders are GPS enabled. In absence of GPS, shared secret based integrity and confidentiality measures should be taken.

VII. Comparison with Existing Classification Methods

In [4], Deng, Li, and Agrawal have discussed the existing secure routing protocols based on attacks those protocols address. For example, SEAD provides robust security against routing attacks targeting the sequence number and routing metrics. Ariadne can defend against routing and wormhole attacks. The ARAN protocol can defend against authentication and repudiation attacks. However, we remark that this kind of

TABLE II
Summary of Scenario Centric Protection Profile. CWP refers to the combination of CONFIDANT, Watchdog and Pathrater.

Scenario	Security Goals	Attacks	Environmental Factors	Protocol
TANET	CONF, AUTH, INTG, AVAL	ACON, AAUTH, AINTG, AAVAL, AENVR	GCAP, BBP, NCH, ELL	SPAAR+CWP
VANET	AUTH, INTG, AVAL	AAUTH, AINTG, AAVAL, AENVR	GCAP, BBA, NCH, ELL	SPAAR, ARAN, SAODV+CWP
AANET	AUTH, INTG, AVAL	AAUTH, AINTG, AAVAL, AENVR	GCAP, BBA, NCH, ELL	SLSP
WSANET	AUTH, INTG, AVAL	AAUTH, AINTG, AAVAL, AENVR	GDIS, BBA, NCL, ELH	N/A
UWANET	AUTH, INTG, AVAL	AAUTH, AINTG, AAVAL, AENVR	GDIS, BBA, NCL, ELH	N/A
CDNET	INTG, AVAL	AINTG, AAVAL, AENVR	GDIS, BBA, NCL, ELH	SAODV
BANET	CONF, AUTH, INTG, AVAL	ACON, AAUTH, AINTG, AAVAL, AENVR	GDIS, BBA, NCL, ELH	N/A
FRANET	CONF, AVAL	ACON, AAVAL, AENVR	GCAP, BBA, NCL, ELH	N/A

approach does not give an overall view of the domain and might not be complete as a classification approach.

In [3], Yih-Chun and Perrig have classified the existing secure routing protocols based on the base routing protocol those protocols follow such as DSR [11], AODV [14] and DSDV [9]. For example: SEAD is based on DSDV. SAODV and ARAN are security extensions to AODV. Protocols like Ariadne and CONFIDANT are based on DSR. This classification approach also lacks in completeness and extendibility.

Fonseca and Festag [20] have classified current secure routing protocols based on security mechanisms used by the protocols such as:

- Asymmetric cryptography: ARAN, SPAAR and SLSP.
- Symmetric cryptography: SEAD and ARIADNE.
- Hybrid approach: SAODV.
- Reputation mechanisms: CONFIDANT and Watchdog-Pathrater.

Now, we compare these classification approaches to our approach in terms of the following metrics:

Completeness: This metric measures to what extent the taxonomy provides a complete overview of secure routing protocols in the ad hoc networks domain. As we have discussed earlier, the existing classification or taxonomic criteria fail to capture the fact that an ad hoc network can manifest itself in different forms. On the contrary, our approach is based on the scenario-centric protection profile, thus covering most of the application scenarios in existing literature.

Usability: This metric is used to measure how simple it is to choose a protocol for an application scenario. Classification mechanisms discussed earlier in this section clearly show that it is very difficult to choose a protocol based on such classification. Although, given an application, an expert might decide whether to use an asymmetric or symmetric approach; absence of environmental factors in the classification criteria hinders the usability of these classification mechanisms. On the other hand, our approach is based on a protection profile where the application scenario is the first consideration making it more usable than the existing ones.

Extendibility: By this metric, we wanted to measure whether this taxonomy could be extended to help the design of new secure routing protocols for ad hoc networks. The protection profile of different scenarios clearly defines the security needs of a specific scenario. By adhering to the protection profile, new secure routing protocols can be designed efficiently. This feature is almost completely absent in the existing classification approaches.

VIII. Conclusion

In this paper, we presented an efficient taxonomy of existing secure routing protocols in different ad hoc network scenarios. We also discussed the suitability of several secure routing protocols for those scenarios. From the taxonomy, it is clear that quick responses are necessary in domains such as body sensor, first responder and underwater ad hoc networks. To facilitate the design of new secure routing protocols, the protection profile of different application scenarios proposed in this work can be a good starting point for researchers.

References

[1] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Mobile Computing and Networking*, 2000, pp. 255–265.

[2] M. G. Zapata, "Secure ad hoc on-demand distance vector routing," *SIGMOBILE Mob. Computer Communication Review*, vol. 6, no. 3, pp. 106–107, 2002.

[3] H. Yih-Chun and A. Perrig, "A survey of secure wireless ad hoc routing," *Security & Privacy, IEEE*, vol. 2, no. 3, pp. 28–39, May-June 2004.

[4] H. Deng, W. Li, and D. Agrawal, "Routing security in wireless ad hoc networks," *Communications Magazine, IEEE*, vol. 40, no. 10, pp. 70–75, Oct 2002.

[5] S. Buchegger and J.-Y. Le Boudec, "Performance analysis of the confidant protocol," in *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, 2002, pp. 226–236.

[6] "Common criteria: The common criteria portal," 2011. [Online]. Available: http://www.commoncriteriaportal.org/

[7] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, "A secure routing protocol for ad hoc networks," in *ICNP '02: Proceedings of the 10th IEEE International Conference on Network Protocols*, Washington, DC, USA, 2002, pp. 78–89.

[8] Y. Hu, D. Johnson, and A. Perrig, "Sead: Secure efficient distance vector routing for mobile wireless ad hoc networks," in *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA 2002), IEEE, Calicoon, NY*, June 2002, pp. 3–13.

[9] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," in *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, 1994, pp. 234–244.

[10] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: a secure on-demand routing protocol for ad hoc networks," *Wireless Networks*, vol. 11, no. 1-2, pp. 21–38, 2005.

[11] D. Johnson, "Routing in ad hoc networks of mobile hosts," in *Proceedings of the Workshop on Mobile Computing Systems and Applications*, Dec 1994, pp. 158 –163.

[12] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The TESLA broadcast authentication protocol," *RSA CryptoBytes*, vol. 5, 2002.

[13] P. Papadimitratos and Z. J. Haas, "Secure link state routing for mobile ad hoc networks," in *SAINT-W '03: Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops)*, 2003, p. 379.

[14] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (aodv) routing," RFC Editor, United States, 2003.

[15] S. Carter and A. Yasinsac., "Secure position aided ad hoc routing protocol," in *Proceedings of the IASTED International Conference on Communications and Computer Networks (CCN02)*, Cambridge, MA, USA, November 2002, pp. 329–334.

[16] Y.-C. Hu, A. Perrig, and D. Johnson, "Packet leashes: a defense against wormhole attacks in wireless networks," *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, vol. 3, pp. 1976–1986, 2003.

[17] M. Al-Shurman, S.-M. Yoo, and S. Park, "Black hole attack in mobile ad hoc networks," in *Proceedings of the 42nd annual Southeast regional conference*, ser. ACM-SE 42. New York, NY, USA: ACM, 2004, pp. 96–97.

[18] M. Khabbazian, H. Mercier, and V. K. Bhargava, "Severity analysis and countermeasure for the wormhole attack in wireless ad hoc networks," *Trans. Wireless. Comm.*, vol. 8, pp. 736–745, February 2009.

[19] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, pp. 382–401, July 1982.

[20] E. Fonseca and A. Festag, "A survey of existing approaches for secure ad hoc routing and their applicability to vanets," NEC Network Laboratories, Heidelberg, Germany, Tech. Rep., March 2006. [Online]. Available: http://www.network-on-wheels.de/downloads/survey_sec_routing_v1-1_cite.pdf

A Holistic, Modular Approach to Infuse Cybersecurity into Undergraduate Computing Degree Programs

Trudy Howles, Carol Romanowski, Sumita Mishra, and Rajendra K. Raj

***Abstract*—In response to societal change and national educational objectives, a holistic, modular approach to Cybersecurity education is presented in this paper. This approach is characterized by a set of reusable, self-contained modules that can be embedded in existing classes in several computing disciplines. The intent is to introduce these modules across computing disciplines, and throughout the undergraduate years to ensure a greater understanding of security issues among diverse computing majors. The ultimate goal is to address the societal need for computing professionals who are educated and experienced in diverse aspects of computing security and information assurance.**

***Index Terms*—Computer Science Education, Computer Security, Curriculum Development, Information Security**

I. Introduction

With the growth and pervasiveness of cyber infrastructure in modern society, secure computing and communicating have become critically important. Applications with critical security requirements include e-commerce, voice/video communications, military operations, secure data management, and financial market transactions. In addition, there is a growing trend toward adding security at various points in the information technology infrastructure, such as embedding in disk drives, processors, trusted system boards, network switching elements, mobile devices, and sensors. All of these points require careful attention to algorithm choice and implementation method, and involve trade-offs between software and hardware.

The development of such systems requires a population of developers who possess the necessary knowledge and design skills. Students must learn fundamental theory and gain practical implementation experience to understand security requirements holistically.

This paper presents an approach to develop and deliver a broad range of concepts and topics associated with security and software design through modules embedded in existing classes in multiple disciplines, including Computer Engineering, Computer Science, Software Engineering, and Cyber Informatics.

These course modules provide sufficient breadth and depth in information assurance concepts, security requirements elicitation and analysis, and specific techniques that are widely used in industry applications. The modules begin with introductory concepts and gradually introduce the necessary theoretical background.

II. A Holistic, Modular Approach

This proposed holistic, modular approach embeds cybersecurity education in computing courses, at the undergraduate level, vertically from freshman to senior year. The course modules in security and information assurance are designed to enhance cybersecurity instruction within the computing disciplines.

Societal change is the impetus behind this approach; the rapid advance of mobile computing devices such as smartphones and tablets, along with the growth of cloud computing, has made cybersecurity a concern of everyone. Issues of cybersecurity and security breaches impact all of society, not just computing professionals [1]. These topics must be incorporated throughout the curriculum, so that students confront these concepts several times during their academic years.

The foundation of the approach is a set of self-contained instructional modules that can be “dropped in” to relevant classes in computing disciplines such as Computer Engineering, Software Engineering, Network Security, Information Technology, and Computer Science. We define a module as a distinct unit of course materials, such as a lab or a teaching component, that can be incorporated into existing courses in the curriculum by an instructor without requiring any course or degree program changes and curricular approval.

Modules cover user, legal, ethical, and technical aspects of cybersecurity in appropriate detail—from introductory

Trudy Howles and Rajendra K. Raj are with the Department of Computer Science, Rochester Institute of Technology, Rochester, NY, 14632 USA (e-mail: {tmh, rkr}@cs.rit.edu).

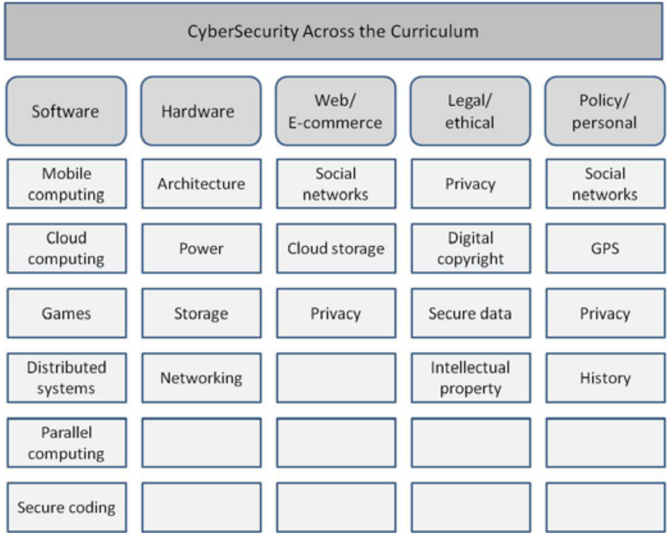
Carol Romanowski is with the Center for Multidisciplinary Studies, Rochester Institute of Technology, Rochester, NY, 14632 USA (e-mail: cjrcms@rit.edu).

Sumita Mishra is with the Department of Networking, Security and Systems Administration, Rochester Institute of Technology, Rochester, NY, 14632 USA (e-mail: sumita.mishra@rit.edu).

material to deeper, more technical concepts—and can be combined into a stand-alone course or sequence of courses in these topics.

The modules are created with a common style that is coordinated across module categories (technical, social, legal, etc.). This coordination extends to the format of exercises, assessments, and other instructional elements. Module developers work with instructional designers to create modules that are self-contained, consistent and can be linked together to form larger blocks of instruction. Each module contains a short introductory section presenting the broad picture of cybersecurity and where each module fits within that overview.

Fig. 1 shows example module categories and possible topics in each category. This figure depicts the long-term vision for how modules and courses can impact a wide cross-disciplinary audience.



CyberSecurity Across the Curriculum

Software	Hardware	Web/ E-commerce	Legal/ ethical	Policy/ personal
Mobile computing	Architecture	Social networks	Privacy	Social networks
Cloud computing	Power	Cloud storage	Digital copyright	GPS
Games	Storage	Privacy	Secure data	Privacy
Distributed systems	Networking		Intellectual property	History
Parallel computing				
Secure coding				

Fig. 1. Module categories and possible topics (Note that topics may span multiple categories, but are shown under one category for convenience.)

Modules are a common pedagogical tool for computing and cybersecurity topics, but mostly used in a single course or set of courses within computing disciplines [2-7]. A recent panel [5] comments on the multidisciplinary nature of security, the need for all computer users to have some security knowledge, and the effectiveness of addressing these concepts in courses holistically rather than focusing on stand-alone classes. Modules have also been used to embed security topics rapidly into an existing information assurance curriculum [7].

III. Sample Course Modules in Computing Disciplines

Some example course modules are presented next, to show the diversity of the modular approach. Additional modules are planned as the effort proceeds.

A. *Module: Security Requirements*

This course module will focus on presenting fundamental security issues and their optimization criteria. A brief overview of standard mechanisms for achieving them will also be visited, such as Input Validation, Access Control, Session Management, Cryptography, Auditing and Logging. This module is to be added to introductory courses for computing and engineering students.

B. *Module: Security of Web Applications*

This set of course modules will concentrate on the specification, design and implementation of specific security requirements introduced in the Security Requirements course module. Possible requirements include Identification and Authentication Requirements; Authorization Requirements; Immunity Requirements; Integrity Requirements; Intrusion Detection Requirements; Non-repudiation Requirements; Privacy Requirements; Security Auditing Requirements; etc. Each course module will focus on one specific security requirement, its optimization criteria, and techniques for achieving and analyzing it. The educational outcome of this set of course modules is to provide student with enough depth in software development, especially web applications, with a strong emphasis on performance analysis of implementation.

C. *Module: Security and Privacy Issues in Network Systems*

Topics include basic cryptographic protocols, key management protocols, Denial-of-Service attacks and defense mechanisms, wireless network security, key management in multicast and wireless sensor networks, Web service security, and network worms. More specifically, the students will learn about the symmetric key cryptographic protocol, public key infrastructure, one-way hash function with data integrity, Diffie-Hellman key exchange, the mechanisms of distributed denial-of-service attacks with possible defense solutions, how to secure wireless LAN, PEAP, and TTLS, how to secure wireless routing, and key management. Students will also discuss how multiple nodes in networks can manage their keys in a secure way for group communications. In addition, security risks and their solutions in Web service including TLS/SSL will be studied, and the spreading patterns of network worms with possible solutions will also be investigated. The students will study these principles, then devise attacks in laboratory exercises or projects, and implement defense solutions in actual systems.

D. *Module: Mathematics for Cryptography*

This course module will focus on discussing the fundamental mathematics used for private and public key encryptions. The purpose of the module is to extract the basics from a comprehensive cryptography class, and introduce security related examples to math, software and hardware design courses. The students are not required to have backgrounds in cryptography or networking. As only a basic knowledge of college-level math is needed, this module will reach a broad audience in addition to computing and engineering students.

E. *Module: Machine Learning for Network Security*

This course module will discuss the application of machine learning algorithms for intrusion detection, alert correlation, and cyber attack prediction. A brief overview of selected machine learning algorithms, such as Bayesian Network, Hidden Markov Models, and Fuzzy Inference will be provided, with case studies of how these algorithms are applied to security problems. The educational outcome of this module is to familiarize students with algorithmic analysis and the implication of machine learning to active cyber defense.

F. Module: Security Issues in a Virtual Environment

This module will introduce the advantages of virtual computing and the security challenges presented by this environment. Full virtualization simulates the underlying computer hardware and enables the software to run without changes. Virtualization enhances the performance of cloud computing systems by optimizing the computing workload and managing the servers more effectively. However, virtualization creates additional layers, hence creating the need for additional security controls and measures. If not managed carefully, the ease of sharing information between systems can be turned into an attack vector very easily. It is challenging to create and maintain the necessary security boundaries due to the dynamic nature of virtualization. Students will learn secure virtualization techniques in the classroom and the concepts will be reinforced through lab exercises.

G. Module: Secure Coding

Following the Software Assurance Curriculum Project Volume II: Undergraduate Course Outlines (from cert.org) as a guide, the initial effort will be a gap analysis to identify the topics missing in current programming courses. Topics will be mapped to existing courses to identify ways in which to integrate new labs or modules to introduce the material. The "CWE/SANS Top 25" most dangerous and critical software vulnerabilities cwe.mitre.org/top25/) [8] will also guide this gap analysis.

Students will study the key vulnerabilities and learn the mitigation techniques for each one. An anonymized archive of previous student project submissions will be used as case studies; students will be challenged to find and correct the vulnerabilities. This is just one hands-on activity designed to help students identify vulnerabilities and weaknesses, and realize the inherent threats in legacy code. Learning modules will address general problems as well as language-specific issues.

To foster a culture of security awareness, students will be required to create a security label (similar to a food nutrition label) for all their course deliverables [9]. These innovative and fun labels serve a dual purpose: to reinforce that programming practices have a continuing impact on the general security of the cyber infrastructure, and that users must become more informed and savvy about what they are purchasing, using and downloading onto their personal devices.

IV. Assessment and Evaluation

For each of the goals, we have developed a set of objectives, each of which has concrete achievable outcomes that will be assessed and evaluated as the effort proceeds. (Following customary pedagogical practice, we use the term *assessment* to mean the gathering of measured data, and the term *evaluation* to mean the analysis of the assessed data.) Findings will be used to refine the learning experience, research activities, and evaluation methodologies.

Table I presents the objectives and intended outcomes of this modular approach to cybersecurity education.

TABLE I
Goal, Objectives, and Outcomes

Goal	Objective	Outcome
1. Embed security and information assurance concepts across the existing university curriculum in computing disciplines.	1.1 Computing majors learn additional Cybersecurity concepts.	1.1.1 Course modules in Cybersecurity are embedded into existing courses taken by computing majors. 1.1.2 Computing majors take additional courses in Cybersecurity.
	1.2 Computing majors apply Cybersecurity concepts in undergraduate capstone projects.	1.2.1 Capstone projects reveal the application of Cybersecurity knowledge.

We plan to use both direct and indirect methods to measure each of the outcomes listed in the rightmost column of Table I. When possible, we will use direct methods, e.g., examination of concrete measureable data, which have greater reliability than indirect methods, e.g., student self-surveys.

V. Current Status of Work in Progress

There are currently three modules that have been piloted to students in the form of short noncredit seminars. The initial modules focus on secure coding, spotlighting the most serious and frequently occurring vulnerabilities. At the conclusion of each seminar, students were asked to provide survey feedback on the content, and their assessment of the relevance of the material. Preliminary feedback indicates that students are interested in the material; this is supported by the strong attendance patterns at the seminars. Student comments indicate that they do recognize the importance of security and safety and are interested in learning more.

Student feedback along with input from industry partners have been helpful in both fine-tuning the content of these early modules, and helping to identify the most critical modules to target next. An additional four modules are currently under development, and are targeted for rollout in the fall of 2011.

VI. Conclusion

This paper outlines efforts to embed cybersecurity modules throughout the computing disciplines, in response to the societal need for computing professionals educated and experienced in computing security. Since security issues affect everyone, not just the core information assurance community, this modular approach ensures that all computing students, regardless of degree program, receive instruction in security concepts.

A different but concurrent effort focuses on the development of similar security modules to disseminate security knowledge to all students, not only computing majors [10]. With the rapid growth in mobile computing devices and diverse types of wireless networking (e.g., 802.11, Bluetooth, 3G, and 4G), computing security has become critical for undergraduate disciplines, just as writing skills are considered important (e.g., Writing Across the Curriculum). Therefore, a conceptual hands-on education in cybersecurity must be provided for as much of the student body as possible. Targeted here are General Education courses offered in non-computing disciplines as well as other relevant classes such as first year experience courses, public policy, criminal justice, psychology, economics, and industrial and systems engineering.

This program enhances our existing academic offerings, and we believe it will attract more students to security-focused majors. The initial positive feedback and increased interest students have shown are encouraging and support this belief. As we develop and deliver more modules, we will continue to survey our students and gather input from our industry partners on the program's direction and effectiveness.

References

[1] B.T. Delp, S. Nuristani, and B. Mitchell, "Cybersecurity: congressional action, public-private partnerships, and education are key to mitigating vulnerabilities," *The CIP Report*, vol. 9, no. 7, pp. 3-18, January 2011.

[2] P. Denning and A. McGettrick, "Recentering computer Science," *Communications of the ACM*, vol. 48, no. 11, pp. 15-19, 2005.

[3] N.Herrmann, J Popyack, B.Char, P. Zoski, C. Cera, R. Lass, A. Nanjappa, "Redesigning Introductory Computer Programming Using Multi-Level Online Modules for a Mixed Audience," *SIGCSE '03*, Reno, Nevada, pp. 196-200, 2003.

[4] S. Sharma, and J Sefchek, "Teaching information security courses: A hands-on approach,"Computers & Security, vol 26, pp. 290-299, 2007.

[5] P. Mullins, J. Wolfe, M. Fry, E. Wynters, W. Calhoun, and R. Montante, "Panel on integrating security concepts into existing computer courses," In *Proceedings of SIGCSE '02*, Covington, Kentucky, pp. 356-366, 2002.

[6] J. Walden, and C. Frank, "Secure software engineering teaching modules," In *Proceedings of the 3rd Annual Conference on Information Security Curriculum Development (InfoSecCD '06)*, Kennesaw, GA, USA, 2006, pp. 19-23.

[7] B. Endicott-Popovsky, and D. Frincke, "A case study in rapid introduction of an information assurance track into a software engineering curriculum," In *Proceedings of the 17th Conference on Software Engineering Education and Training*, pp. 118-123, 2004.

[8] Mitre Corporation. "CWE - 2010 CWE/SANS Top 25 Most Dangerous Software Errors," [Online]. 2010. Available: http://cwe.mitre.org/top25.

[9] Aspect Security. "Application Security Facts," [Online]. 2011. Available: http://www.aspectsecurity.com/SecurityFacts.

[10] R. K. Raj, S. Mishra, T. Howles, C. Romanowski, "Cybersecurity as General Education," In *Proceedings of 15th Colloquium for Information Systems Security Education (CISSE '11)*, Fairborn, OH, USA, June 2011, to be published.

AUTHOR BIOGRAPHIES

Dr. Frank Adelstein received his B.S. Eng. in computer engineering from the University of Michigan in 1988, and his M.S. and Ph.D. in computer science from The Ohio State University in 1990 and 1995. He is currently a Senior Staff Scientist at ATC-NY in Ithaca, NY. He coauthored a book on mobile computing and is the vice-chair for the Digital Forensics Research Workshop. Computer security, including digital forensics, and networking are his primary areas of focus.

Kenneth Beck is a M.S. student in Computer Science program at the Rochester Institute of Technology.

Stephen Brueckner is Technical Director of Cyber Security at ATC-NY in Ithaca, NY. He has led and collaborated on research and development efforts in the fields of intrusion detection, online training, virtualization, digital forensics, secure collaboration, service-oriented architectures, and P2P networking. Formerly a consulting geologist, with degrees from Cornell and Drexel Universities, he received his Master's degree in computer science from Oregon's Portland State University in 2004.

Vincent J. Buccigrossi III is a B.S. student in the Information Security and Forensics Program at the Rochester Institute of Technology.

Dr. Jason W. Clark is a security researcher who studies privacy and security issues of computer-based systems. Mr. Clark has a Bachelor of Science in Information Technology from Syracuse University as well as a Master of Science in Information Technology from Rensselaer Polytechnic Institute (RPI) and a second master's degree in Computer Forensics from George Mason University (GMU). Mr. Clark is currently working on his Ph.D. at GMU in the field of information technology/security with a focus on securing anonymizing network systems. He also works full-time as a lead information security analyst.

Asif Dipon is a M.S. student in Computer Science program at the Rochester Institute of Technology.

Matthew Donovan is a Principal Scientist at ATC-NY. His research interests include Microsoft Windows internals, virtualization, and application security. He received his B.S. in physics from SUNY Oswego and his M.S. in computer science from Portland State University.

Manish Gupta is a Ph.D. candidate at State University of New York at Buffalo. He also works full-time as an information security professional at a northeast US based bank. He received an MBA from SUNY-Buffalo (USA) and a bachelor's degree in mechanical engineering from I.E.T, Lucknow (India). He has more than a decade of industry experience in information systems, policies and technologies. He has published 3 books in the area of information security and assurance. He has published more than 50 research articles in leading journals, conference proceedings and books including DSS, ACM Transactions, IEEE and JOEUC. He serves on editorial boards of 7 International Journals and has served in program committees of several international conferences. He holds several professional designations including CISSP, CISA, CISM, ISSPCS and PMP.

Kevin Highley graduated from Morehead State University. He worked on "Runway control using smart sensor networks" with Dr. Panja.

Dr. Diane T. Hooie is a Senior Advisor with the Energy Delivery Technologies Division of the Project Management Center at the Department of Energy's National Energy Technology Laboratory. She has over 35 years of experience converting new ideas and innovative technologies from the concept stage through production and to profitable marketable products. Her current responsibilities include implementing the Cyber Security for Energy Delivery Systems Program for the office of Electricity Delivery and Energy Reliability as well as developing technical collaborations with non-traditional DOE customers, such as the Department of Homeland Security and Department of Defense, and developing international programs including Russia, Kazakhstan, Egypt, and Japan in the clean energy technology areas including clean coal, electricity, turbines, fuel cells, hybrids, and fuels. She received her BS in Ceramic Engineering from Ohio State University, MS in Management from Rensselaer Polytechnic Institute, and a PhD in Engineering from California Coast University. During her career, Dr. Hooie has received many awards and honors and has over 100 publications and presentations, including two books and one encyclopedia article, pertaining to fuel cells, fuels, and turbines. In 1998, she was selected "Woman of the Year," and the highest honor, "Person of Distinction," for the Federal Government.

Dr. Trudy Howles is an Associate Professor in computer science. Her current research areas include secure software and privacy preserving data mining. Dr. Howles is a Senior Member of the American Society for Quality, an Associate Editor for the Software Quality Professional, and is active with the Software Assurance Workforce Education and Training Working Group, which is co-sponsored by organizations in the DoD, DHS and NIST. She has also been an active researcher on student educational and retention strategies with presentations at IEEE and ACM conferences, and publications in ACM Inroads, Software Quality Professional, and Computer Science Education.

Mohammad Iftekhar Husain is a PhD candidate at the CSE department of University at Buffalo (SUNY-Buffalo). His research interests include, but are not limited to: cloud computing security, social network analysis-based network security, steganography, and economics of information security. Mohammad has completed his master's degree in 2008 from the same department. He has a BS in Computer Science from Yamagata University, Japan. Mohammad has received multiple international scholarships including the Rotary Ambassadorial Scholarship and the Monbusho Scholarship for academic achievement.

Dr. Hajime Inoue is a Principal Scientist at ATC-NY. His research interests focus on combining programming language implementation with computer security and machine learning. He received his doctorate in Computer Science from the University of New Mexico.

Dr. Hilmi Güneş Kayacık is a postdoctoral fellow at Carleton University, School of Computer Science. His research interests include machine learning and its applications to network analysis, intrusion detection and data leak prevention.

Nathan LaFontaine is a B.S. student in the Computer Science program at the Rochester Institute of Technology.

Justin Lewis is a B.S. student in the Computer Science program at the Rochester Institute of Technology.

Priyanka Meharia is a graduating Ph.D. student at Gatton College of Business of University of Kentucky.

Stefanie Milstead is a M.S. student in Computer Science program at the Rochester Institute of Technology.

Dr. Sumita Mishra is an Assistant Professor in the Department of Networking, Security and Systems Administration. Her current research interests are in security for sensor and smart grid networks. Dr. Mishra is working with researchers at Syracuse University, Virginia Tech, and CCNY on the development of open specification of neighborhood area networks and wireless grids. She has published over 45 papers in international journals and conferences. She has also served as a reviewer for numerous journals and conferences, including IEEE Transactions on Vehicular Technology, IEEE Transactions on Communications, IEEE Transactions on Mobile Computing, IEEE INFOCOM, ICC, GLOBECOM and ACM MOBICOM.

Yin Pan, Ph.D. is an Associate Professor in the Department of Networking, Security and Systems Administration at RIT. Dr. Pan holds four US patents in the areas of Network Quality of Services, Voice over IP and Artificial Intelligence. Since 2004, Dr. Pan has been actively involved in the network security area, especially in IT security audits and computer forensics. Her current research involves information security auditing, social networking security, and computer forensics. She has over 35 publications in journals and conferences.

Dr. Biswajit Panja is an Assistant Professor of Computer Science at Morehead State University. He earned his PhD from the University of Missouri-Rolla.

Onur Polatcan, M.S. is a Network Engineer in the Dep. Assistant Secretary for Information and Technology, VA Medical Center in Indianapolis, IN. His research interests include social engineering, information retrieval, text computations, and analysis of cybercrimes.

Dr. Rajendra K. Raj is a Professor in Computer Science whose current research interests include secure software systems and secure data management in distributed and cloud settings. He has also focused on CS education and is currently involved in curriculum development and assessment (K-12, undergraduate, and graduate levels). Dr. Raj has served as an external program evaluator for several computing programs across the U.S. Prior to RIT, he served as a Vice President in Information Technology at Morgan Stanley & Co, New York, where he developed and managed secure distributed data infrastructures used worldwide to build financial software applications

Preeti Raman is a Ph.D. candidate at Carleton University, School of Computer Science. Her research interests include data leak prevention, bio-inspired computing, anomaly detection, self-healing software, and software security.

Dr. H. R. 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, as well as the co-Editor- in -Chief of Information Systems Frontiers.

Dr. Leon Reznik is a Professor of Computer Science at the Rochester Institute of Technology, New York, U.S.A. He received his B.S./M.S. 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 U.S.A. since 2002. Professor 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), and "Advancing Computing and Information Sciences" (Cary Graphic Arts Press, 2005). Dr. Reznik's research concentrates on the study and development of intelligent computing systems for controls, sensor networks and systems as well as cyber security applications.

Dr. Carol Romanowski is an Assistant Professor in the Center for Multidisciplinary Studies (CMS). Her current research interests are in applied data mining in engineering design, product lifecycle management, and critical infrastructure. Dr. Romanowski is co-PI on an Urban Area Security Initiative grant focusing on critical infrastructure analysis tool development for Tier 2 (midsize) urban areas. She is on the editorial board of the International Journal of Data Mining, Modelling, and Management and has published in IEEE Transactions, Computers and Operations Research, and the Journal of Computing and Information Science in Engineering.

Dr. Raj Sharman is an Associate Professor in the Management Science and Systems Department at SUNY, Buffalo, NY. He received his B. Tech and M. Tech degree from IIT Bombay, India and his M.S degree in Industrial Engineering and PhD in Computer Science from Louisiana State University. His research streams include Information Assurance, and Disaster Response Management, Business Value of IT, Decision Support Systems, Conceptual Modeling and Distributed Computing. His papers have been published in a number of national and international journals. He is also the recipient of several grants from the university as well as external agencies.

Holden Silvia is a B.S. student in the Computer Science program at the Rochester Institute of Technology.

Dr. Anil Somayaji is an Associate Professor in the School of Computer Science at Carleton University in Ottawa, Canada and Associate Director of the Carleton Computer Security Lab. He received a B.S. (1994) degree in mathematics from the Massachusetts Institute of Technology and the Ph.D. (2002) degree in computer science from the University of New Mexico. He has served as the program committee chair of the New Security Paradigms Workshop (2008 and 2009) and has served on the program committees of ACM CCS, USENIX Security, and RAID, among others. His research interests include computer security, operating systems, complex adaptive systems, and artificial life.

Dr. Ramalingam Sridhar received a B.E. (Honors) degree in Electrical and Electronics Engineering from Guindy Engineering College, University of Madras in 1980, MS and PhD in Electrical and Computer Engineering from Washington State University in 1983 and 1987 respectively. Since 1987, he has been with the University at Buffalo, The State University of New York where he is an Associate Professor in the Department of Computer Science and Engineering. His research interests are in embedded cloud technologies, security, wireless and sensor network security, secure architectures, reliable deep submicron VLSI systems, power aware embedded solutions, clocking & synchronization, and memory circuits & architecture. He was an IEEE Circuits & Systems Distinguished Lecturer. He has served as Program Chair and General Chair of ASIC/SoC Conference and has served in the editorial board of many journals and technical committee of numerous conferences in wireless systems and VLSI.

Dr. Angelos Stavrou is an Assistant Professor in the Department of Information and Software Engineering and a member of the Center for Secure Information Systems at George Mason University, Fairfax, Virginia. He received his M.Sc. in Electrical Engineering, M. Phil. and Ph.D. (with distinction) in Computer Science, all from Columbia University. He also holds a M.Sc. in Theoretical Computer Science from the University of Athens, and a B.Sc. in Physics with distinction from the University of Patras, Greece. His current research interests include security and reliability for distributed systems, security principles for virtualization, and anonymity with a focus on building and deploying large-scale systems. He is a member of the ACM, IEEE, and USENIX.

Dr. Peter Stephenson is in the Department of Computing at Norwich University. He teaches digital forensics and digital investigation. He is a member of the Distinguished Faculty in the School of Graduate Studies and has over 45 years experience in various technology and information assurance fields. He has written or contributed to 16 books and several hundred articles in major national and international trade publications and technical/scientific journals. He holds CISSP, CISM and FICAF certifications and is a special member of the Vidocq Society.

Richard D. Walter is a retired Prison Psychologist and Crime Assessment Expert. He is one of the co-founders of The Vidocq Society and one of the persons who is highlighted in the recently released book, "The Murder Room" written by Michael Capuzzo. He is said to be one of the creators of "modern" criminal profiling. Presently, he is a consultant and lecturer to various law enforcement agencies and professional groups.

INDEX OF AUTHORS