Précis: Research on Techniques and Tools for Computer Security: The COAST Project and Laboratory

Director: *Eugene H. Spafford*
Department of Computer Sciences
1398 Computer Science Building
Purdue University
West Lafayette, IN 47907–1398
spaf@cs.purdue.edu
+1 317 494-7825 (office)
+1 317 494-0739 (fax)

Last revision: 12 September 1996

Abstract

The goal of the **COAST** project is to establish a long-term research program exploring new approaches to computer security and computer system management. The principal focus will be on techniques and tools for common, non-trusted (in the DoD TCSEC, "Orange Book" sense) systems, especially legacy systems in wide-spread use. We anticipate that the work we do in this area will also aid in the development of future systems. A secondary goal of **COAST** is to advance the state of education in computer security, and includes the establishment of a dedicated, dual-use laboratory. We intend for our research to be broad-based and to draw from experience and expertise in other areas of computing such as software engineering, language design, operating systems, computer networking, and databases. It will also draw on expertise in related non-computer areas, as appropriate to advance our goals.

COAST is supported through grants and gifts, and by donations of software and equipment. Some work is supported through research contracts. Our sponsors get early access to **COAST** products, software and publications, as well as unique opportunities for research collaboration. **COAST** (see section 3 for a full list) also include special access to the (see section 4) conducting the research. Our sponsors help to focus our research: we expect continuing, direct feedback from them on our research directions and results.

By combining support from many different sources, we help leverage the contributions of individual sponsors, resulting in significantly greater levels of effort. We expect our research to lend new insights into the development and use of tools for enhanced computer security and operational management. By involving our sponsors in our work, we improve its applicability and enhance technology transfer. We expect that our research results will have a significant impact on the current state of practice as our results are brought into use.

1 Introduction

It is clear that computer security is an area of increasing, major concern and that the computing community is facing some severe challenges related to security. Incidents related to disclosure of information, wide-scale computer breakins, and the exponential growth in the number of computer viruses being written and discovered all indicate an increasing threat to effective use of computing resources.[1]

The recent report *Computers at Risk*[32], forcefully outlines several critical security problems facing computer users. Our increasing reliance on computers for critical applications poses increasing temptation for unauthorized criminal and terroristic activity. Our increased connectivity provided by new network technologies simply amplifies the threat. Additionally, the increased use of computer databases, and the increasing use of parallel and distributed processing all add new variables that have not, as yet, been adequately examined. Initiatives that link together computing systems from around the world and that provide access to more users, such as the National Research and Engineering Network (NREN), National Information Infrastructure (NII) and High Performance Computing (HPC) initiatives, will likely add to the potential for security problems.

As was noted in a recent Office of Technology Assessment report:

> Information networks are changing the way we do business, educate our children, deliver government services, and dispense health care. Information technologies are intruding in our lives in both positive and negative ways.... As businesses and governments become more dependent on networked computer information, the more vulnerable we are to having private and confidential information fall into the hands of the unintended or unauthorized person.... Otherwise, concerns for the security and privacy of networked information may limit the usefulness and acceptance of the global information infrastructure.[19, Forward]

The urgency of the problem is more directly stated in the recent report of the Joint Security Commission[2]:

> Nowhere is this more apparent than in the area of information systems and networks. The Commission considers the security of information systems and networks to be the major security challenge of this decade and possibly the next century and believes that there is insufficient awareness of the grave risks we face in this area. The nation's increased dependence upon the reliable performance of the massive information systems and networks that control the basic functions of our infrastructure carries with it an increased security risk. Never has information been more accessible or more vulnerable. This vulnerability applies not only to government information but also to the information held by private citizens and institutions. We have neither come to grips with the enormity of the problem nor devoted the resources necessary to understand fully, much less rise to, the challenge. ... Protecting the confidentiality, integrity, and availability of the nation's information systems and information assets—both public and private—must be among our highest national priorities.[12, p. 2]

It is interesting to note that this conclusion is independent of whether or not there is concern for protection against directed "information warfare."

Computer security, as an area of specialization, is difficult to accurately define. Even professionals working in the area of computer security have difficulty agreeing on an exact definition that appropriately

[1] See, for instance, [20], and the on-going series of advisories from response teams such as the CERT Coordination Center, Department of Energy CIAC, NASA NASIRC, and DISA ASSIST.

[2] This commission was composed of personnel of the U.S. Department of Defense, Central Intelligence Agency, and Department of Energy. The charter of the Commission was to evaluate the current state of security in their agencies, and to suggest needs and directions.

encompasses the field. Part of the reason that security is difficult to describe is because it draws heavily upon so many areas of computing. In at least one sense, it seems closely related to software engineering — computer security is devoted to ensuring that software and hardware meet their specifications and requirements when used in a potentially hostile environment. Computer security thus includes issues in computer system specification, verification, testing, validation, safety, and reliability. However, security encompasses much more than these issues, including topics in (at the least) operating systems design, architectural design, information security, risk analysis and prediction, database organization, encryption and coding, formal models of computation, fault tolerance, network and protocol design, supportive interface design, government regulation and policy, managerial decisions, security awareness, and education.

The difficulty in defining computing security is also reflected in the scattered and underdeveloped educational and research programs in the area. While other fields of computing research have well-defined bodies of educational literature, the field of computer security is largely defined by short chapters in textbooks on operating systems, data communications, and databases. The field currently has only a few widely-circulated archival journals with refereed articles on computer security topics: e.g., *Computers & Security*, *Cryptologia*, and the relatively new *Journal of Computer Security*.

Few universities or colleges offer in-depth education in computer security. As of mid-1995, there were only three dedicated computer security research labs in degree-granting departments at universities in the United States[3] When computer security courses are taught, relatively few textbooks on computer security are in use, and several of the most commonly used ones are principally devoted to cryptography (e.g., [6]). Research in academia is being done by a limited number of faculty at scattered locations working with a few students. What research is being done, in academia or commercially, is primarily oriented towards limited military requirements.

This lack of visibility, training, and coordinated research efforts has led to a shortage of practitioners trained in *practical* computing security, and to a critical shortage of academic faculty prepared to offer advanced instruction in this area. This contributes to a lack of consideration of security issues when new computer systems are being designed, thus placing those new systems at risk. As technology propels us into a future where global networks of communicating, multi-vendor computer systems are commonplace, the lack of universally-accepted social norms and laws will lead to difficulties that only well-designed computer security tools and techniques may prevent. To design those tools and train that workforce, we need an experienced, well-educated cadre of faculty.

Research in computer security-related issues has usually been conducted under a number of different rubrics reflecting its cross-disciplinary nature. Work in areas such as computer architecture, operating systems, data communications, database systems, and software engineering has included questions of computer security. Despite advances in all these areas, most direct security-related research in the last few decades has been largely directed towards only a few selected topics. For instance, most of the systems-oriented research done to date has been in support of formal trust models for multi-level secure machines employed in military settings, including compartmented-mode workstations. The results of this research is usually of little use in "real-world" computing environments. Furthermore, as more computer users seek to use COTS (commercial, off-the-shelf) components, we will need better protection methods built in to the common system.

In particular, considerable research in computer security methods and protocols over the last few decades

[3] In addition to **COAST**, these are the programs at U.C. Davis and George Mason University. The **COAST** group at Purdue is perhaps the largest of the three, and is certainly the one with the broadest focus.

has largely been focused on issues of theoretical models of secure systems, multi-level systems, covert channels, statistical intrusion detection systems, and communications security issues (e.g., cryptography). Insufficient research has been focused on the development of tools for improving general security, audit techniques, availability models, network security, computer forensics, countering malicious software (e.g., computer viruses and worms), policy formation, and integrity methods. Perhaps this is because work in all these areas requires an interdisciplinary approach and few researchers and research groups have both the breadth and depth of expertise necessary to conduct such investigation. It may also be because the traditional focus of research has primarily been in military systems' computer security, which focuses on issues of confidentiality [17, 18]. Thus, there has been less support for research in the area of designing security tools and techniques for everyday use on commercial and educational computing platforms than on military systems.

2 COAST Goals and Recent Research

The **COAST** (**C**omputer **O**perations, **A**udit, and **S**ecurity **T**echnology) Project is directed to computer security research on general and practical tools and techniques. The focus of this research is on typical computing environments — systems without multi-level requirements, and without formal levels of "trust." In particular, our short-term research is directed to developing approaches of increasing the security of existing systems without severely impacting their usability. Our goal is to explore how to increase confidence in existing systems in a cost-effective and user-friendly manner. Our long-term research is directed to how to integrate better security mechanisms into common computing platforms.

COAST was established as a formal structure in the spring of 1992 by Eugene Spafford after several years of independent research work. The idea for **COAST** was to bring together expertise of several faculty, share resources in computer security research, and provide a unified approach to our local research and education efforts in this vital area. Most of the recent research in **COAST** has been related to his efforts and projects, but others within the department have been involved in research, and several more have expressed commitment to more fully examining the security implications of their areas (provided that outside interest warrants it). As outside support and participation increases, and as we develop more infrastructure locally, we expect **COAST** to grow to include many more people than currently involved.

Our vision for the long-term goals of **COAST** is to have it be:

- A world-recognized center of research excellence. We intend to be known for our research into methods of practical computer security technology, including computer incident response, system management and network security technologies. We expect most of our research to be based on the real needs of the community, as conveyed to us through interactions with our sponsors and the user population. **COAST** is already known world-wide, and we intend to build on our existing reputation.
- A renowned source of educational and training materials in computer and network security. We intend to produce materials for use in computer security training, both for in-service training and for academic use. This includes traditional materials such as texts and lab materials, but will also include leading-edge technology as embodied in hypermedia and distance-learning methodologies.
- An on-going source of quality graduates with training in computer and network security. We expect our undergraduate and graduate students to receive a broad-based and comprehensive education that will

give them a solid foundation for work in computer security, computer systems, and communication networks.

- A resource center for research. We hope to build a comprehensive collection of documents, references, tools, hardware, software, testbeds, and other resources necessary for comprehensive research and experimentation in various areas of computer security. We hope to make the **COAST** Laboratory (formally dedicated on March 11, 1996) a significant, widely-available resource for visiting scholars, sponsor personnel, and **COAST** researchers.
- A resource center for independent evaluation of products. We expect to be able to provide unbiased, comprehensive testing and evaluation of security tools for computers and networks. By providing detailed test results to sponsors, vendors, and the user population, we believe we will help improve the overall state of information system security.
- A resource center for information dissemination to the non-technical community. There is a significant need for sources of information for the press and public that is unbiased by commercial interest or government agenda. We expect to continue to be known and consulted as one such source. **COAST** personnel have been quoted on issues of computer security and computer crime over 150 times in the last four years, including quotations in the *New York Times*, *Newsweek*, the *Wall St. Journal*, *Newsweek*, *NPR Radio*, *ABC Radio*, *Scientific American*, *Science* and more.
- A source of useful tools for system management and security. Although not a primary focus of the **COAST**, we expect that we will produce new tools and protocols as useful byproducts of our research activity that will be of wide-spread applicability to the community at large. The **COAST** on-line archive is already acknowledged as the single largest and most comprehensive security repository on the Internet.

Significantly, **COAST** has already made progress in each of these areas.

Since 1987, Professor Spafford and his students have been exploring issues in *practical* computer security. Their work has included widely-cited work in analysis of malicious code including viruses (e.g., [26, 25, 29, 30, 31]). In 1991, Spaf coauthored the award-winning book *Practical* UNIX *and Internet Security*[8], now considered the standard reference in the field. He has also been involved in work on static audit and analysis tools. An initial result of this work was the COPS[7] security audit tool for UNIX systems, used worldwide on tens of thousands of computer systems. This tool runs on several dozen varieties of UNIX, and detects scores of configuration and management problems that may lead to security problems. It is still viewed by many as the standard of its type.

Another tool Spaf designed, OPUS, [28, 27] has been experimented with as one of our **COAST** projects. This system will help prevent users from choosing weak or obvious passwords, yet it will not add to the administrative burdens of the system administrator, nor will it require the megabytes of disk storage that would be required of a comprehensive dictionary-based screening mechanism of similar power. Furthermore, an explicit design goal was to prevent the tool from being useful as a means of breaking passwords. Experience has shown that over 80% of attacks from the outside of networked systems are based on exploitation of weak passwords[4], so we expect OPUS to also enjoy widespread application.

Other research in the center is being conducted on integrity monitoring methods for virus protection, intrusion detection, and change management. One of the first results of this work was a portable integrity

[4] According to unpublished reports from the CERT Coordination Center and other incident response centers.

checking tool to search for unauthorized changes to files, as might be made by the addition of a trapdoor, logic bomb or virus: *Tripwire*, (currently in release, version 1.2[5])[13, 15, 14]. We have also expended some effort on developing an efficient scanner tool that can be configured to search for indicated bit patterns (e.g., a virus signature); this reconfigurable virus-scanning tool was described in [16].

A former graduate student, Steve Chapin, supported under a NASA Graduate Research Fellowship was working with us on scheduling support for fully autonomous distributed systems [4, 5].[6] This research was devoted to development of mechanisms and protocols that support cooperation of machines in a network without forcing them to give up local control of resources or scheduling autonomy. A direct result of this research is support for distributed scheduling that cannot result in (global) denial of service because each individual machine may control allocation of its own internal resources at all times. Future research into availability methods is under active consideration.

Two current projects involve audit and supervision concerns. One project, IDIOT,[7] was funded by the Department of Defense. It employs new methods of specifying and detecting anomalous behavior and exceptional events. Unlike a simple intrusion detection mechanism, this work is intend to serve as a general-purpose monitor for all forms of notable behavior without placing undue load on the system being monitored. The IDIOT protoype was developed as part of Sandeep Kumar's Ph.D. work. It is now being enhanced for release to **COAST** sponsors. A second student's Ph.D. project is examining how to monitor host and network activity in real time and react to designated traffic. The focus of Mark Crosbie's work is how to scale the monitoring in a cost-effective manner.

Under examination for future development within **COAST** are a COPS-like tool for auditing network security and configuration; next-generation network firewall technology on ATM networks; new methods of testing software to discover potential security vulnerabilities; methods of secure distribution of vulnerability information; analysis of security flaws; development of mechanisms to counter denial of service attacks; development of incident response and investigation tools; formalisms for testing firewall devices; and developing a classification of audit information to separate anomaly detection, accounting, and record-keeping functions.

We have recently established the largest and most comprehensive on-line archive[8] of computer security tools and documents on the Internet. This archive mirrors tools and documentation held on scores of sites around the Internet, and is organized according to topic. The server has already been mirrored to sites throughout the world, and regularly sees downloads in excess of 100 megabytes per day. The archive also includes a gopher and a WWW interface, too.

Overall, the **COAST** philosophy is to conduct coordinated, cooperative research in the area of security tools and methods. Uniquely, a primary facet of our work is to investigate methods of developing and employing practical security tools and techniques that are of little or no threat to systems. This research is thus directed towards development of simple approaches and reconfigurable tools that do not require secrecy or extreme privilege to be effective, and that cannot be easily used for system penetration, nor be easily altered to introduce "trojan horses." This work should also result in new insights into the development of secure software.

COAST research will continue to focus on practical application of the techniques developed, with

[5] ftp://coast.cs.purdue.edu/pub/COAST/Tripwire/tripwire-1.2.tar.Z

[6] He has since graduated and joined the faculty at Kent State University.

[7] Intrusion Detection In Our Time.

[8] ftp://coast.cs.purdue.edu/pub

experimental trials in existing systems, at Purdue and elsewhere, to validate the methods used. We hope that sponsors will participate in these trials, to provide us with the kind of real-world constraints and environments that are so seldom present in the experiments conducted by other researchers.

As noted above, another explicit philosophy of the **COAST** group is to provide students with the opportunity to be involved with computer security issues. By providing research and educational opportunities, we hope to encourage good students to specialize in the area. We also hope to use these results to develop new curricular materials in computer security. By encouraging participation of **COAST** sponsors, we further intend to widen the base and reach of the computer security educational materials that we expect to produce. The result should be students better equipped to undertake a career in industry or academe with computer security as their specialty.

3 Sponsors

Companies and government agencies may become sponsors of the **COAST** project and laboratory. Sponsors will help determine the direction of **COAST** research, through regular feedback, and through an advisory panel. Sponsors will receive early releases of prototype software and our reports, whenever practical. We also anticipate other interactions with sponsors, including placing students for internships, personnel exchanges, and tutorials. The nature of the research proposed for **COAST** is such that active collaboration of sponsors will be sought to validate and refine **COAST** products.

Sponsors will be recognized in **COAST** publications and products (unless they request otherwise). Sponsors will also have preferential access to faculty, staff and students involved in **COAST** projects. Sponsors will have opportunity to provide significant input into the direction of our research. At the least, we anticipate the following as significant benefits of sponsorship:

- Sponsors will gain early access to new computer security technology.
- Sponsors will gain a means to influence and guide cutting-edge research in computer security at an early stage.
- Sponsors will gain preferential access to students working in **COAST**. This may give an “inside track” to hiring talented new employees with experience in computer security research.
- Sponsors will have a means of helping influence security in commercial products through **COAST** development, and through contact with other **COAST** sponsors.
- Sponsors will gain visibility in **COAST** activities, and publications.
- Sponsors gain preferential access for product evaluation, bug fixes, comparison testing, etc. done by **COAST** personnel.
- Sponsors may gain early access to warnings of security problems and trends identified by project personnel.
- Sponsors will get early access to prototype tools and technology to be used as a basis for further development for both internal use and for use by sponsor “clients.”

- Sponsors gain access to a pool of expertise that can be tapped to help deal with unexpected problems that might not be covered by in-house expertise.
- Sponsors will help improve the general posture of security in the computing community by supporting **COAST** efforts.
- Sponsors will be able to obtain educational materials on security.
- Sponsors will gain early access to technical reports, software prototypes, and other products of **COAST**.
- Sponsors will get automatic access to seminars, workshops, classes, and other activities to be sponsored by **COAST**.

We are interested in joining forces with additional sponsors for **COAST**. We prefer a steady, multi-year commitment of support from each sponsor rather than one-time, large grants; most of the projects planned will be multi-year in scope. Single donations for specific projects and purposes are still appropriate, however, and constitute sponsorship.

Companies and agencies may also become contributors to (instead of sponsors of) **COAST**. This will entail a smaller donation or gift to the project, and will not result in our same levels of support or disclosure as for sponsors. However, we will seek to involve and support **COAST** contributors more closely than non-contributors. Support at the level of $50,000 or above per year constitutes full sponsorship.

Sponsorship of **COAST** may be in one of four ways:

1. **By unrestricted grant**. A sponsor may make an unrestricted gift of money to Purdue University, on behalf of the **COAST** Project and its director. This may then be used to support worthy students, fund publication and software distribution, send **COAST** personnel to appropriate conferences, and purchase equipment and software for the lab. This is the prefered form of support: gift/grant money is subject to a much lower overhead charge by Purdue than other forms of support. Grants and gifts of money and/or equipment may have certain Federal and state tax advantages for the donor, too.

   (NB: As of June 1, 1996, support of COAST no longer qualifies for a discount in becoming a full corporate partner in the CS Department's corporate partner program[9].)

2. **By directed contract**. A sponsor may specifically fund one or more particular projects within **COAST**, or fund the project as a whole. This involves negotiating a contract with the Purdue Research Foundation for specific deliverables (e.g., reports or software), and would need to include coverage of salaries, equipment, travel, supplies, secretarial expenses, and overhead costs; in general, about 1/3 of every research contract goes to these fixed overhead costs.

   Because of the basic philosophy behind **COAST**, and because of the university mandate for publication of research results, offers of contracts for proprietary or classified research are not solicited.

3. **By fellowship**. A sponsor may wish to directly offer multi-year, continuing scholarship support to graduate students (or undergraduates) working on **COAST**-related projects. These fellowships may be named and designated largely as the sponsor wishes. These would be funded as either gifts to **COAST** or as awards to the principal investigator, and designated specifically as fellowships for

[9]http://www.cs.purdue.edu/homes/asn/CPP.html

students involved in **COAST** research. Sponsorships involving a periodic on-site (sponsor location) internship are possible and encouraged.

4. **By equipment and software donation**. One of the goals of the project is to develop and maintain a state-of-the-art laboratory for computer security research and education. This requires an *on-going* procurement of equipment and software. Sponsors may donate *appropriate* equipment and materials to help stock and maintain this lab. Any equipment so donated will largely define the "baseline" systems used for development and distribution of **COAST** products.

   We note, however, departmental policy that assesses a fixed maintenance cost per software architecture supported. This covers installation and support of third-party software, local operational security, backups, printing support, and other necessary tasks. The **COAST** group is charged at a rate of $15,000 to $25,000 per year for any software architecture not currently present in quantity within the department. This means that a donation of equipment might be reluctantly declined unless it includes sufficient money for support.

Prospective sponsors are invited to contact Gene Spafford for further information about opportunities to sponsor **COAST** projects and participate in the project and its results.

Current sponsors of **COAST** or **COAST** projects are:

DARPA
Hewlett-Packard Company
MITRE, Inc.
Schlumberger, Limited
U.S. Sprint
Sun Microsystems
U.S. National Security Agency

Current supporters of **COAST** include:

Haystack Labs
Internet Security Systems, Inc
Security Dynamics Corporation
Thomson Consumer Electronics
U.S. Air Force Information Warfare Center
Xerox PARC

Past Sponsors and Contributors include:

Baseline Software, Inc.
Enigma Logic, Inc.
FSA Corporation
Hughes Research Laboratories/Hughes Aircraft Company
IBM
InternetOne, Inc.
Nortel (formerly Bell Northern Research)
Raxco
Trident Data Systems

4 Personnel and Resources

The Department of Computer Sciences at Purdue University is the nation's oldest computer science department (founded in 1962), and has a long tradition of research and service in computer systems, software engineering and computer security. Computer security, in particular, has a long history at Purdue. A graduate course in computer security has been taught here for fifteen years, after being started by Dorothy Denning when she was at Purdue.

The Department of Computer Sciences is considered by many to be in the top dozen programs overall in computer sciences in the country. The department includes dedicated labs for several major projects and centers, including the SoftLab project, the Software Engineering Research Center (SERC), and the Parallal and Distributed Systems (PADS) group.

The department offers degrees at all levels, including an active PhD program graduating almost a dozen new PhDs in computer science each year from an outstanding graduate student population. The program currently enrolls about 450 undergraduates and 150 full-time graduate students. The current faculty of more than three dozen researchers includes internationally-known individuals with interests that include networking, computer systems, databases, algorithms and data structures, user interfaces, graphics and visualization, parallel computation, computational theory, software engineering, performance evaluation, artificial intelligence, and robotics. The faculty represent, literally, dozens of international journal editor positions, and more than a score official positions in professional and technical organizations including the ACM, the IEEE and IEEE Computer Society, SIAM, and the Computing Research Association.

Research funding in the department continues to average around $5 million per year. In some recent figures, the department was second in the nation in funding through the NSF Division of Computer and Computation Research, and had twice the research funding (from NSF) of any other Big Ten school. The department is also notable for the substantial support it receives from industry sources.

Principal Investigator and Director

Gene Spafford attended the School of Information and Computer Sciences (now the College of Computing) at Georgia Institute of Technology, holding both a Georgia Tech President's Fellowship and a National Science Foundation Graduate Fellowship. He received the Ph.D. in 1986 for his design and implementation of the original Clouds reliable, distributed operating system kernel, and for his contributions as one of the original members of the Clouds design team. Next, Dr. Spafford spent a year and a half as a research scientist with the Software Engineering Center at Georgia Tech. His duties there included serving as a principal software engineer with the Mothra software testing project.

In 1987, Professor Spafford joined the academic faculty of the Department of Computer Sciences at Purdue University, where he now holds the rank of associate professor. At Purdue, he has taught courses in operating systems, compiler and language design, computer security, computer architecture, software engineering, networking and data communications, and issues of ethics and professional responsibility. Professor Spafford has twice been cited in the last five years as one of the 10 best undergraduate instructors in the School of Science.

At Purdue he has conducted research on methods of increasing the reliability of computer systems, and the consequences of computer failures. This has included work with computer security, software testing and debugging, and issues of liability and professional ethics. Spaf is also an active member of the Software Engineering Research Center (SERC) at Purdue — an NSF-founded Industry-University Cooperative Research Center.

Professor Spafford is currently on the advisory and editorial boards of the *International Journal of Computer and Software Engineering*, the *Virus Bulletin*, the *Journal of Artificial Life*, the journal *Network Security*, *Computers & Security*, and the *Journal of Information Systems Security*. He has written extensively in the field of computer security, including coauthoring an award-winning book on UNIX Security ([8]) and a widely-cited book on computer viruses ([30]). He has published over 100 papers and reports on his research. He has also spoken internationally at panels, conferences, symposia, and colloquia on these issues.

Dr. Spafford is a member of the Association for Computing Machinery (ACM) and has served as chairman of both that organization's Self-Assessment Committee and of its ISEF Awards Committee, as a charter member of the Technical Standards Committee, and as a member of the Committee on U.S. Public Policy (USACM). He is a Senior Member of the IEEE and of the Computer Society of the IEEE; he was recently named as a member of the Computer Society's *Golden Core*. He is also a member of the Usenix Association. He is a member of the advisory board of the National Research Center on Computing and Society. Spaf was a founder of the Purdue Computer Emergency Response Team (a FIRST member organization[10]); Spaf has served as a member of the international FIRST steering committee. Spaf is a member and immediate past chair of the IFIP Technical Committee 11 Working Group on Network Security (WG 11.4), and a member of Working Group 11.8, Information Security Education and Training.

Other information may be found on Professor Spafford's WWW homepage[11].

Other Personnel

As of fall 1996, twenty-two students are working on security-related projects with **COAST** faculty. This group includes two Fulbright Fellows and students holding several other awards; recent graduates include an NSF Graduate Fellow and a GM Fellow. Many of these students are members of international honor societies such as Phi Beta Kappa, Phi Beta Phi, Upsilon Pi Epsilon and Tau Beta Pi, and most currently hold competitive fellowships or assistantships.

Several Purdue faculty members are also interested in computer security topics. What follows are brief descriptions of a few of the faculty who have expressed strong interest in collaborating on projects as part of the **COAST** effort, or who have been actively involved in **COAST** research. Their interests and areas of expertise help provide additional depth and potential to the **COAST** effort.

Mikhail Atallah

Mikhail J. Atallah received the Ph.D. in Electrical Engineering and Computer Science from Johns Hopkins University, Baltimore, MD, in 1982. He then joined the faculty of Purdue University, West Lafayette, Indiana, where he is currently Professor of Computer Science. In 1985 he received a Presidential Young Investigator award from the National Science Foundation. In the summer of 1988 he was a Visiting Scientist at the NASA Ames Research Center (RIACS Institute, Center for Advanced Architectures).

[10] FIRST is the Forum of Incident Response and Security Teams.
[11] http://www.cs.purdue.edu/people/spaf

Dr. Atallah is a member of the ACM, a Senior Member of the IEEE, and a member of the Society for Industrial and Applied Mathematics (SIAM). He currently serves on the editorial boards of the journals *Computational Geometry: Theory & Applications*, *Information Processing Letters*, *Int. J. on Computational Geometry & Applications*, *J. of Parallel and Distributed Computing*, *Methods of Logic in Computer Science*, *Parallel Processing Letters*, *SIAM J. on Computing*, and of the Wiley *Handbook of Parallel and Distributed Computing*.

Professor Atallah's main research interests are the design and analysis of combinatorial algorithms, parallel computation, and computational geometry. He has coauthored over 85 technical papers in these areas, e.g.[2, 1] He is currently working with Professor Spafford on the application of special-case pattern matching algorithms with temporal characteristics to intrusion detection systems.

Professor Atallah is also interested in audit trail reduction. Audit trail files have a special structure, with many repetitions of a certain kind, so the challenge is to (i) Do better for compressing them than using generally known data compression techniques such as Lempel-Ziv, and (ii) How to search for a pattern in a compressed file without having to "decompress."

More information may be found on Professor Atallah's WWW homepage.[12]

Carla Brodley

Carla Brodley received the Ph.D. in Computer Science in 1994 from the University of Massachusetts. She then joined the faculty of Purdue University, West Lafayette, Indiana, where she is currently an assistant professor in the School of Electrical and Computer Engineering.

Professor Brodley's main research interests are in in the areas of machine learning, pattern recognition and knowledge discovery in databases. She has worked in the areas of anomaly detection (detecting observations that differ from the normal cases), classifier formation (forming a discrete classification system to perform prediction), and feature selection (determining which measurable features are predictive of the output of interest). She has applied techniques from these areas to problems from a variety of fields including computer vision, remote sensing and medical diagnosis.

Professor Brodley is a member of AAAI and IEEE. She currently serves on the editorial board of the journal *Intelligent Data Analysis*, a new (1997) electronic journal.

More information may be found on Professor Brodley's WWW homepage[13]

Tony Hosking

Professor Hosking received the B.Sc. from the University of Adelaide, the M.Sc. from the University of Waikato, and the Ph.D. in 1995 from the University of Massachusetts. His dissertation work in persistent programming languages demonstrated that the fundamental mechanisms of persistence can be supported efficiently on stock hardware, even for very fine-grained data.

Having joined Purdue as an assistant professor in 1995, Professor Hosking is now exploring the advantages of compiler support for high-performance persistent and distributed programming languages and

[12] http://www.cs.purdue.edu/people/mja

[13] http://mow.ecn.purdue.edu/~brodley/

systems [9, 10, 11]. These interests mesh naturally with issues of security in distributed systems, especially for environments that support the distribution and mobility of very fine-grained objects. Maintaining security in such environments can be particularly difficult to achieve without compromising performance.

More information may be found at Professor Hosking's WWW homepage.[14]

John T. Korb

Dr. Korb received his Ph.D. from the University of Arizona in 1979, where he participated in the design and implementation of the Icon programming language. He has held both academic and research positions with the Computer Sciences faculty at Purdue in the past dozen years. He is currently the Director of Research Facilities for the department. His research interests include operating systems, programming languages, networks, and user interfaces. He has lectured widely on Internetworking and the X Window System, and is involved with many of the research projects at Purdue.

Supervising the departmental computing facilities, Dr. Korb's responsibilities make him acutely aware of the current problems with practical computer system security, both from technical and administrative points of view. He is keenly interested in performance monitoring and testing tools to detect and correct security problems, including intruders and viruses. Additionally, he is interested in the development of new techniques and tools for managing a multi-vendor, networked installation: currently, he is responsible for almost 400 machines running over a dozen different operating systems.

More information may be found on Dr. Korb's WWW homepage.[15]

Vincent F. Russo

Professor Russo received his Ph.D. in computer science from the University of Illinois at Urbana-Champaign in 1990. His dissertation work was in object-oriented operating systems; he designed the *Choices* operating system there. Since joining the faculty at Purdue as an assistant professor, Vince has been actively involved in research in the areas of distributed operating system design and object-oriented systems.

Currently, Vince is directing research on the *Renaissance* project [21]. One goal of this project is to provide the ability for unrelated applications written in a variety of programming languages to obtain access to objects distributed throughout a network of machines and implemented in a variety of manners [22]. The approach being used is to create a first-class description of an object's interface that has a runtime realization and is independent of any programming language or system. A translator is then provided to convert this description into a variety of languages and systems.

An example of this would be providing object-oriented access to files resident on an NFS file server. These descriptions will be used to generate interfaces to remote objects dynamically at runtime. They will also allow flexible but complete interface specification providing both increased encapsulation and documentation. The intent is to make it possible to map non-object-oriented substrates into systems by viewing objects as servers and using the client–server model common to many non-object-oriented distributed applications.

[14] http://www.cs.purdue.edu/people/hosking
[15] http://www.cs.purdue.edu/people/jtk

A central issue in any distributed object model, including this one, is security: deciding who has access to an object and with what permissions. Dr. Russo and one of his graduate students are investigating how to provide this security in an efficient and safe manner. He is also interested in using the *Renaissance* system as an operating system testbed for further experiments in security enhancements to operating systems.

More information may be found on Professor Russo's WWW homepage.[16]

H. J. Siegel

H. J. Siegel received his Ph.D. degree in 1977 from the Department of Electrical Engineering and Computer Science at Princeton University. In 1976, he joined the faculty of the School of Electrical Engineering at Purdue University, where he is a Professor and Coordinator of the Parallel Processing Laboratory.

Professor Siegel is a Fellow of the IEEE, was a Coeditor-in-Chief of the *Journal of Parallel and Distributed Computing*, and is on the Editorial Boards of the *IEEE Transactions on Parallel and Distributed Systems* and the *IEEE Transactions on Computers*. From 1979 to 1982, he was an *IEEE Computer Society Distinguished Visitor*, giving invited lectures across the country about his research. He is a member of the Eta Kappa Nu electrical engineering honorary society and the Sigma Xi science honorary society.

Prof. Siegel has coauthored over 150 technical papers and has presented his work at conferences in the USA, Europe, Japan, and Israel. He wrote the book *Interconnection Networks for Large-Scale Parallel Processing* (second edition 1990) and has coedited five other volumes.

His current research focuses on interconnection networks, heterogeneous computing, and the use and design of the PASM reconfigurable parallel computer system (a prototype of which is supporting active experimentation). He is interested in exploring some of the security challenges posed by parallel machines, and the possible benefit of applying parallel technology to security concerns in more traditional architectures.

More information may be found on Professor Siegel's WWW homepage.[17]

Samuel S. Wagstaff, Jr.

Professor Wagstaff obtained his Ph.D. in mathematics at Cornell University in 1970. He has worked in Computational Number Theory ever since then. For the past dozen years his research has focused on Public Key Cryptography, especially on the computational complexity of factoring integers. While at the University of Georgia in 1981–83, he helped to design and build a special processor for factoring large integers. In 1983, he joined the faculty of the Department of Computer Sciences at Purdue University. He has taught a course in Cryptography and Data Security almost every year since then, as well as teaching courses in several other areas.

Dr. Wagstaff is an Associate Editor of the journal *Advances in the Theory of Computation and Computational Mathematics*. He is coauthor of a popular book on factorization of numbers of the form $b^n \pm 1$ [3]. He has given dozens of lectures around the world on Computational Number Theory, and has published extensively in this and related areas (e.g., [23, 24]).

Sam's interests include most areas of computer security. He hopes to work with the **COAST** Project on matters related to cryptography. His interests mesh especially well with **COAST** projects involving autho-

[16] http://www.cs.purdue.edu/people/russo
[17] http://dynamo.ecn.purdue.edu/Faculty/siegel.html

rization, applying zero-knowledge proofs techniques, and integrity checking methods. One current project, being investigated jointly with Professor Spafford, involves finding mechanisms for the safe distribution of sensitive software patches and upgrades.

More information may be found on Professor Wagstaff's WWW homepage.[18]

Michal Young

Professor Young received the Ph.D. in computer sciences from the University of California, Irvine, in 1989 and joined Purdue as an assistant professor. In 1991 he was honored by selection as an NSF Presidential Young Investigator.[19]

Michal's research interests focus on improving the dependability of concurrent (distributed, parallel, and multi-threaded) software. He is currently directing research on modular state-space analysis of concurrent software systems (using process algebra to avoid combinatorial explosion when analyzing the interactions among many processes [33]), and concurrency control and scheduling for hard real-time systems. He is also pursuing research in software development environments and integration of testing, analysis, and verification techniques [34].

One of Professor Young's recent research projects involves generation of state-space analysis tools for combinations of (possibly domain-specific) descriptions of processes. A typical use of generated tools would be detection of races, deadlocks, and other coordination problems in an evolving or heterogeneous system. Another potential use is in analyzing vulnerabilities by modeling a system design together with attacks of various kinds. This could be used to discover and analyze security weaknesses in real-time and highly concurrent computing systems.

An area of interest for future **COAST** research, shared with Professor Spafford, is the modeling of availability of computational resources and developing methods to ensure continued availability. Denial of service is a significant problem in security, and research in this area is sorely needed.

More information may be found on Professor Young's WWW homepage.[20]

5 For More Information

The current PostScript version of this document. is available for anonymous ftp[21]. A version[22] of that document is also available in Adobe Acrobat PDF. A hypertext, somewhat expanded version of this précis is available as http://www.cs.purdue.edu/coast/precis/precis.html. There is also a WWW homepage[23] for **COAST**. Professor Spafford may also be contacted directly for further information about **COAST**.

[18] http://www.cs.purdue.edu/people/ssw

[19] Support of **COAST** research involving Professor Young may qualify for special matching funding from the National Science Foundation under this award.

[20] http://www.cs.purdue.edu/people/young

[21] ftp://coast.cs.purdue.edu/pub/COAST/What_is_COAST.ps

[22] ftp://coast.cs.purdue.edu/pub/COAST/What_is_COAST.pdf

[23] http://www.cs.purdue.edu/coast/coast.html

References

[1] M. J. Atallah, P. Jacquet, and W. Szpankowski. A probabilistic approach to pattern matching with mismatches. *Random Structures and Algorithms*, 1994. to appear.

[2] M. J. Atallah, C. Lock, D. C. Marinescu, H. J. Siegel, and T. L. Casavant. Models and algorithms for co-scheduling compute-intensive tasks on a network of workstations. *Journal of Parallel and Distributed Computing*, 16:319–327, 1992.

[3] John Brillhart, D. H. Lehmer, J. L. Selfridge, Bryant Tuckerman, and S. S. Wagstaff, Jr. *Factorizations of* $b^n \pm 1$*, b = 2, 3, 5, 6, 7, 10, 11, 12 up to high powers*, volume 22 of *Contemporary Mathematics*. American Mathematical Society, Providence, RI, second edition, 1988.

[4] Steven J. Chapin and Eugene H. Spafford. Constructing distributed schedulers using the messiahs interface language. In *27th Hawaii International Conference on Systems and Software (HICSS)*, Wailea-Maui, Hawaii, 1994. Available as http://www.cs.purdue.edu/homes/spaf/tech-reps/9383.ps.

[5] Steven J. Chapin and Eugene H. Spafford. Support for security in distributed systems using messiahs. In *Proceedings of the National Computer Security Conference*, pages 339–447, Oct 1994. Available as http://www.cs.purdue.edu/homes/spaf/tech-reps/sjc-ncsc.ps.

[6] Dorothy E. R. Denning. *Cryptography and Data Security*. Addison-Wesley, Reading, MA, 1983.

[7] Daniel Farmer and Eugene H. Spafford. The COPS security checker system. In *Proceedings of the Summer Conference*, Berkeley, CA, June 1990. Usenix Association. Also available as http://www.cs.purdue.edu/homes/spaf/tech-reps/993.ps.

[8] Simson Garfinkel and Gene Spafford. *Practical Unix & Internet Security*. O'Reilly & Associates, Inc., Sebastapol, CA, second edition, 1996. Summary information available as http://www.ora.com/item/pus2.html.

[9] Antony L. Hosking, Eric Brown, and J. Eliot B. Moss. Update logging for persistent programming languages: A comparative performance evaluation. In *Proceedings of the Nineteenth International Conference on Very Large Data Bases*, pages 429–440, Dublin, Ireland, August 1993. Morgan Kaufmann. Also available as ftp://ftp.cs.umass.edu/pub/osl/papers/vldb93.ps.Z.

[10] Antony L. Hosking and J. Eliot B. Moss. Object fault handling for persistent programming languages: A performance evaluation. In *Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications*, pages 288–303, Washington, DC, October 1993. *ACM SIGPLAN Notices 27*, 11 (November 1993). Also available as ftp://ftp.cs.umass.edu/pub/osl/papers/oopsla93.ps.Z.

[11] Antony L. Hosking and J. Eliot B. Moss. Protection traps and alternatives for memory management of an object-oriented language. In *Proceedings of the Fourteenth ACM Symposium on Operating Systems Principles*, pages 106–119, Asheville, North Carolina, December 1993. *ACM Operating Systems Review 27*, 5 (December 1993). Also available as ftp://ftp.cs.umass.edu/pub/osl/papers/sosp93.ps.Z.

[12] Joint Security Commission. Report of the joint commission. Technical report, U.S. Government, 1994.

[13] Gene H. Kim and Eugene H. Spafford. Monitoring file system integrity on unix platforms. *InfoSecurity News*, 4(4):21–22, July 1993. Available as http://www.cs.purdue.edu/homes/spaf/tech-reps/gkim.

[14] Gene H. Kim and Eugene H. Spafford. Experiences with tripwire: Using integrity checkers for intrusion detection. In *Systems Administration, Networking and Security Conference III*. Usenix, April 1994. Available as http://www.cs.purdue.edu/homes/spaf/tech-reps/9412.ps.

[15] Gene H. Kim and Eugene H. Spafford. Writing, supporting, and evaluating tripwire: A publically available security tool. In *Proceedings of the Usenix Applications Development Symposium*, Berkeley, CA, 1994. Usenix. Available as http://www.cs.purdue.edu/homes/spaf/tech-reps/9419.ps.

[16] Sandeep Kumar and Eugene H. Spafford. A generic virus scanner in C++. In *Proceedings of the 8th Computer Security Applications Conference*, pages 210–219, Los Alamitos CA, December 1992. ACM and IEEE, IEEE Press. Also available as http://www.cs.purdue.edu/homes/spaf/tech-reps/9262.ps.

[17] National Computer Security Center. Trusted computer system evaluation criteria. Technical Report DoD 5200.28-STD, U.S. Department of Defense, 1985.

[18] National Computer Security Center. Computer security subsystem interpretation of trusted computer system evaluation criteria. Technical Report NCSC-TG-009, U.S. Department of Defense, 1988.

[19] Information security and privacy in network environments. U.S. Office of Technology Assessment report, September 1994.

[20] Richard Power. Current and future danger. Technical report, Computer Security Institute, San Francisco, CA, 1995.

[21] Vincent F. Russo. Process scheduling in the Renaissance object-oriented multiprocessor operating system. In *Proceedings of SEDMS II*. Usenix, 1991.

[22] Vincent F. Russo and Patrick A. Muckelbauer. Distributed object interoperability via a network type system. In *Proceedings of the Workshop on Object Orientation in Operating Systems*. IEEE, 1991.

[23] J. W. Smith and S. S. Wagstaff, Jr. How to crack an RSA cryptosystem. *Congressus Numerantium*, 40:367–373, 1983.

[24] J. W. Smith and S. S. Wagstaff, Jr. Methods of factoring large integers. In D. V. Chudnovsky, G. V. Chudnovsky, H. Cohn, and M. B. Nathanson, editors, *Number Theory, New York, 1984-85*, volume 1240 of *Lecture Notes in Mathematics*, pages 225–234. Springer-Verlag, 1987.

[25] Eugene H. Spafford. An analysis of the internet worm. In C. Ghezzi and J. A. McDermid, editors, *Proceedings of the 2nd European Software Engineering Conference*, number 387 in Lecture Notes in Computer Science, pages 446–468. Springer-Verlag, September 1989. Also available as http://www.cs.purdue.edu/homes/spaf/tech-reps/933.ps.

[26] Eugene H. Spafford. The Internet Worm: Crisis and aftermath. *Communications of the ACM*, 32(6):678–687, June 1989. An expanded version is available as http://www.cs.purdue.edu/homes/spaf/tech-reps/823.ps.

[27] Eugene H. Spafford. Observing reusable password choices. In *Proceedings of the 3rd UNIX Security Symposium*, pages 299–312, Berkeley, CA, September 1992. Usenix Association. Also available as http://www.cs.purdue.edu/homes/spaf/tech-reps/9149.ps.

[28] Eugene H. Spafford. OPUS: Preventing weak password choices. *Computers & Security*, 11(3):273–278, May 1992. Available as http://www.cs.purdue.edu/homes/spaf/tech-reps/9128.ps.

[29] Eugene H. Spafford. Computer viruses as artificial life. *Journal of Artificial Life*, 1(3):249–265, 1994. Also available as http://www.cs.purdue.edu/homes/spaf/tech-reps/ALife.ps.

[30] Eugene H. Spafford, Kathleen A. Heaphy, and David J. Ferbrache. *Computer Viruses: Dealing with Electronic Vandalism and Programmed Threats*. ADAPSO, Arlington, VA, 1989.

[31] Eugene H. Spafford and Stephen A. Weeber. Software forensics: Can we track code to its authors? In *Proceedings of the 15th National Computer Security Conference*, pages 641–650, Washington, DC, October 1992. National Institute of Standards and National Computer Security Center. Also available as http://www.cs.purdue.edu/homes/spaf/tech-reps/9210.ps.

[32] National Research Council System Security Study Committee. *Computers at Risk: Safe Computing in the Information Age*. National Academy Press, 1991.

[33] Wei Jen Yeh and Michal Young. Compositional reachability analysis using process algebra. In *Proceedings of the 4th Symposium on Testing, Analysis and Verification*, pages 49–59. ACM SIGSOFT, October 1991.

[34] Michal Young and Richard N. Taylor. Rethinking the taxonomy of fault detection techniques. In *Proceedings of the 11th International Conference on Software Engineering*, pages 53–62, May 1989.