Proceedings of the Fourth Workshop on Future Directions in Computer Misuse and Anomaly Detection

Monterey, California
November 12 - 14, 1996

Edited by:

Karl Levitt
Christopher Wee

Sponsored by:

National Security Agency, R-23
Department of Defense

Office of Research and Development
Central Intelligence Agency

Air Force Information Warfare Center
United States Air Force
Department of Defense

Naval Postgraduate School

Department of Computer Science
University of California, Davis

CMAD IV

COMPUTER MISUSE & ANOMALY DETECTION

SESSION SUMMARIES

MONTEREY, CALIFORNIA
NOVEMBER 12-14, 1996

Prepared by Karl Levitt, Jennifer Sharps, Christopher Wee, Todd Heberlein, Gene Spafford, Becky Bace and Susan Gragg. Publication assistance by Mary Brown.

Program Committee

Karl Levitt (UC Davis), chair

Cynthia Irvine (Naval Postgraduate School), co-chair

Jim Anderson (James Anderson Co.)

Susan Gragg (Office of Research & Development)

Kathleen Jackson (Los Alamos National Laboratories)

Don Marks (Office of INFOSEC Computer Science)

J.F. Mergen (BBN)

Peter Neumann (SRI International)

Jennifer Sharps (Office of Research & Development)

Steve Smaha (Haystack Laboratories)

Eugene Spafford (Purdue University)

Christopher Wee (UC Davis)

Kevin Ziese (US Air Force)

Acknowledgements

The Program Committee expresses its thanks to ORD, UC Davis, Department of Computer Science, and Kevin Ziese and the Air Force Information Warfare Center, and Prof. Charles Martel and the Department of Computer Science, UC Davis, for sponsoring this workshop.

Special thanks go to all the experts and participants in the workshop.

We appreciate the help of the following graduate students, postgraduate researchers, and alumni of the UC Davis Computer Security Laboratory who worked as scribes throughout the workshop:

Steven Cheung	James Hoagland	Steven Samorodin
Rick Crawford	James Pace	Raymond Yip
Jeremy Frank	Nick Puketza	Dan Zerkle

Finally, we wish to thank Allison Mitchell, Debbie Chadwick, Michele Fulton, and Meshell Hays (UC Davis), who handled the myriad of details associated with arrangements and administration for the workshop.

Table of Contents

Executive Summary

Karl Levitt

The Fourth Workshop on Future Directions in Computer Misuse and Anomaly Detection (CMAD IV) was held on November 12-14, 1996 at the Naval Postgraduate School in Monterey, California. In its short history, CMAD has been a forum for discussing the problems associated with intrusion detection, including the promotion of new research ideas, and the application of the technology to real-world problems. Experts are invited from varied parts of the community including academicians directly involved in intrusion detection research and other complementary research fields, government researchers and practitioners with security needs, industry developers with real-world experience in intrusion detection products, and graduate students.

The meteoric rise of the Internet in global commerce and its widespread use in communication makes solving information security problems over wide-area networks of paramount importance. The building blocks of the Internet are components that are intrinsically not secure, and given their complexity, it is not possible to build secure components. Furthermore, for most applications, the definition of security is ambiguous, and will remain so until security policies are created for the myriad uses of large systems. Hence, it will be necessary to retrofit flexible security measures into existing systems. One solution is intrusion detection, which involves the detection of activity that threatens a system. The focus of the intrusion detection community has been on *detection,* but the broader problem of response in the face of a detected incident is also being considered.

The CMAD IV workshop was organized around seven sessions held over three days. A summary of the sessions follows:

1. **Policy-Driven Intrusion Detection and the Insider Threat**: The goal of this session was to explore techniques to detect insiders that exceed their privileges. It is necessary to characterize what activity is permitted of an insider or the organization's *policy.* The experience of the financial community in detecting fraud (e.g., involving credit cards) was considered a benchmark solution to the insider problem. Key issues of computer security include what features to audit, and how to formulate a security policy governing the actions of insiders, if more than anomaly detection is desired. It was concluded that application-level auditing (e.g., the accesses to a database management system) is necessary to detect subtle misuse.

2. **Intrusion Detection Technology for Small-Scale Systems**: Generally, this arena is where most successes of intrusion detection have occurred; many products that detect intrusions into hosts or small networks are now available. Most depend on attack signatures to detect known attacks or variants of known attacks, but some also use the techniques of anomaly detection to detect activity inconsistent with profiles of use. There is a pressing need to develop a methodology for evaluating these (and forthcoming) products. According to users, most of the products suffer from a proliferation of false positives and are incapable of detecting new attacks.

3. **New Attacks and New Twists on Existing Attacks**: The session presented a complete review of existing attacks and attack methods, but focused on new attacks currently experienced and drew predictions about new waves of attacks. For example, denial-of-service attacks are becoming more prevalent. The participants concluded that almost all components that comprise the Internet are vulnerable. Different domains are subject to attack, including the WWW, cellular mobile phones, and a whole host of services, such as Corba. Models of vulnerabilities were presented as a step towards predicting future attacks. Large-scale attacks that could threaten a large network were also discussed and are the basis for the concepts explored in the next session.

4. **Intrusion Detection in the Large**: Although perhaps suitable as building blocks, small-scale systems presented in session (2) are an incomplete solution to a system that applies intrusion detection throughout a large network. There are many problems associated with such a system, including scalability in data collection, reporting, and interpretation. A number of new system concepts are being explored that distribute data collection, detect suspicious activity, correlate activities (data fusion), and decide and carry out appropriate responses. Important issues are *trust* among the components involved in the above processes and coping with missing or compromised data.

5. **New Environments for Intrusion Detection**: Most of the work on intrusion detection has been driven by attacks on Unix and on the Unix implementation of protocol hierarchy. There is a clear need to consider application to the NT class of operating systems, an effort that is just receiving attention. Coping with data-driven attacks, where a malicious program is hidden in data, is an issue. A "system" approach to security was also discussed, in which intrusion detection is one component along with prevention methods. A proxy firewall that provides filtering, based on access control rules (type enforcement) and auditing, was proposed for study of the tradeoffs between detection and prevention. A separate discussion focused on a standard format for audit data (and other data associated with an intrusion detection system), that suggested the possibility of a self-defining format that captures the syntax and semantics of a component's reporting data.

6. **Tools for Investigative Support**: This session explored the use of intrusion detection tools to deal with an incident: assembling evidence that points back to the guilty party, prosecuting them, and using the experience gained in the process to guide the development of next generation tools. Real-life incidents were presented to focus the discussion. Examples of the use of existing tools in dealing with incidents were presented, and in light of ever changing privacy laws, legal issues underlying the use of detection tools were discussed. A critical problem, one central to the intrusion detection problem, is how to cope with the large amount of data while searching an audit log.

7. **New Ideas**: This session focused on new directions for the field, including research, new products, and application needs. In the area of research, intrusion detection needs to work at the enterprise level, which will involve issues of policy formulation, correlation and aggregation, response, recovery, and the synergistic integration of prevention and detection methods. In the area of products, the current products need to be evaluated according to some accepted criteria and move beyond Unix-centered tools. Furthermore,

the tools need to be affordable and maintainable. Overall, we must include ways to protect legacy systems to prevent performance loss.

Session 1: Policy-Driven Intrusion Detection and the Insider Threat

***Moderator and Session Editor**: Jennifer Sharps*

Presenters: Jim Anderson, Marvin Schaefer, Sal Stolfo, Dai Vu, Raymond Yip, Gene Schultz

The Policy-Driven Intrusion Detection and the Insider Threat Audit Panel was assembled to elucidate the problem of detecting computer misuse by the authorized user.

Jim Anderson, James P. Anderson Co., started the panel presentations with a statement of the problem. Anderson defined misuse as an excess of (authorized) access and distinguished the difference between misuse and intrusions.

Marv Schaefer, Arca Systems Inc., then discussed how traditional audit collection and analysis may provide insufficient information for misuse detection. Anderson and Schaefer agreed that since misuse is contextual, there is a vital need for increased application auditing.

Sal Stolfo, Columbia University, described his experience with establishing a fraud detection system for a consortium of banks. Even though rampant misuse is present in financial systems, the community's unwillingness to share information makes misuse detection extremely difficult.

Dai Vu, Lockheed Martin, presented a model that maintains a separate policy database which defines "normal" vs. "abnormal" usage as a function of who is using the system and in what role (rather than the traditional "hardwing" of policy into detection systems). Such a model could support per-application use where users are granted specific functions in an application depending on their identity and specific role.

Raymond Yip, UC Davis, discussed the difficulty of detecting misuse within database systems. Yip's emphasis was exploring methods to mitigate the inference threat. The threat of inference stems from users being able to obtain information from databases without direct retrieval by submitting a set of innocuous queries that logically infer sensitive data.

Gene Schultz, SRI Consulting, wrapped up the panel by relaying a number of interesting examples of insider misuse to illustrate that the threat is very real. Schultz stated we should not rely on traditional computer science alone to detect computer misuse. Schultz suggested using computer-based detection in conjunction with psychological and situational models to lessen the insider threat.

Last, the panel created a list of items to be addressed:

- Resolve current vulnerabilities; don't just patch them
- Increase the level and quality of application auditing to feed into detection systems
- Develop systems that can accommodate manager-approved transactions and/or encapsulated transactions to prevent system damage

- Focus more research on the subtle misuse problem rather than intrusion detection
- Realize firewalls are only part of a correct security solution not a panacea
- Prevent alien software from becoming a virtual "insider" on a system
- Develop clear, concise, workable policies to be implemented on systems
- Raise insider threat awareness and accountability for organizations
- Acknowledge and understand privacy issues for system monitoring

Session 2: Intrusion Detection Technology for Small-Scale Systems

***Moderator and Session Editor**: Karl Levitt (substitute for Tim Grance)*

Presenters: Steve Smaha, Kathleen Jackson, Phil Porras, Calvin Ko, Steve Hofmeyr, Aziz Mounji, Richard Lippmann

The goal of the session was to consider the wealth of production quality and research intrusion detection systems (IDS) intended to monitor small-scale systems that include single hosts or small to moderate size networks. The networks of interest here are associated with a laboratory or office, up to a network of a large company, but administered at a single site. The systems discussed in this session would be components for a large-scale intrusion detection system, as discussed in a later session. The session commenced with presentations, as summarized below, followed by a productive discussion on how to evaluate current products, directions for future products, and research questions on small-scale intrusion detection systems.

Steve Smaha presented the work that was ongoing at Haystack Laboratories, in particular the *Stalker* line of systems. Different from many experimental systems, Stalker is intended for production use, which includes full-time availability, robust operation (acceptable false negative and false positive rates) and accountability. Stalker is a centralized monitoring system that provides four main functions: 1) audit control, 2) tracer and browser of audit records, 3) a misuse (signature-based) detector, and 4) storage management. It is capable of detecting attacks known as doorknob rattling, rdist flaw exploitation to gain root access, ICMP bombs, various Trojan horses, unauthorized NFS mounts, portmapper problems, the presence of a password sniffer (ed: not clear how this is detected), and sweeps (e.g., by SATAN). All of these attacks are detected through built-in signatures that are collectively pre-processed to reduce their matching against observed activity.

An interesting and important issue is the complexity of signatures to detect these kinds of attacks, and whether there is a need for a special language (other than regular expressions) to facilitate the creation of signatures. Steve did not anticipate the need for such a language, as the signatures seem to be short and simple. Although early versions of Steve's systems, such as the experimental *haystack* system, used statistical anomaly detection, it is Steve's opinion that statistical methods do not work well. He is also skeptical of AI-based systems, to perform situation assessment or decide on automated responses, for example. However, Stalker does include a real-time automated response capability, for example, to configure a router to cut and exclude subsequent connections from sources Stalker concludes are attack sites.

Kathleen Jackson described NADIR (Network Anomaly Detection and Intrusion Reporter), which although an experimental system, has been deployed to monitor Los Alamos' large (9000 users) network. Using audit logs as a data source, NADIR works mostly offline to:

- Get an overall impression of current conditions on the network
- Spot and evaluate probes

- Learn about activities of the users of the network
- Make it clear to users that it is monitoring them for suspicious activity, in essence serving as a deterrent.

NADIR uses both statistical methods and built-in signatures. The results of its deployment are very promising: few false positives and no serious incidents were detected by manual review of the audit logs. NADIR fully occupies five workstations, but the cost in terms of human experts to review its reports is low.

Phil Porras, as part of his graduate work at UC Santa Barbara, developed the *State Transition Analysis System* (STAT), which ultimately became the experimental Solaris-based system D-STAT. The system uses signatures and is intended to detect known attacks. There is an interesting twist in the way signatures are processes which gives the system the capability to detect variants of known attacks, in principle. Most signature-based systems would require a separate signature for each variant. Since the number of variants can be unbounded, these systems would be nearly useless against an attacker with the capability of launching an attack not covered by a signature. D-STAT keeps a record of *interesting* states, where certain events can cause state transitions. By specifying the possible events by pre-conditions and post-conditions, it is possible to determine, if given a particular state of the system, whether an event will cause a transition to an interesting state – for example, yield the attacker root access. D-STAT has been shown to detect several dozen attacks, including all attacks known at the time of the presentation.

Calvin Ko presented the *Specification-Based Intrusion Detection* system, in which activity inconsistent with user-supplied specifications for a system is flagged as a possible attack. Calvin has written specifications for most of the Unix privileged programs, because an attacker can exploit vulnerabilities in these programs causing unauthorized actions, such as changing permissions on *shell* to run as root. Since the specifications are not specific to an attack, the method can be used to detect unknown attacks, in principle – provided the specifications cover the actions of the attack. The current specifications indicate files that the privileged programs can open, files that can have permissions changed, programs that privileged programs can execute, and privileged ports they can listen to. These properties are sufficient to detect all known attacks that exploit vulnerabilities in privileged programs. Calvin has produced and made available an implementation of his technique on Solaris 2.5 that uses BMS audit reports as the data source. Although it has not been implemented, the specifications seem to be sufficiently simple to allow synthesis by profiling the privileged programs in the style of anomaly detection systems such as NIDES. The specification-based method is being extended to services such as DNS and routing, and again can detect all of the known attacks against these services in addition to some hypothesized attacks.

Steve Hofmeyer's work, under the leadership of Stephanie Forrest, was inspired by the ideas of immunology, in particular the concept of *self*. An implementation of a service at a site will exhibit a behavior that is specific to the site. For example, the *sendmail* system might make patterns of system calls that are specific to the site it runs on, perhaps because it has been tailored to meet the needs of the site. An interesting twist on this

idea, inspired by diversity in biological systems, is to have privileged programs (those that can be exploited by attackers to gain unauthorized access to a system) contain differences across a network to prevent an entire network from being successfully attacked by a single event stream. Steve's work has shown that relatively short patterns (containing only six system calls) can detect known attacks against many of the privileged programs, e.g., sendmail. The patterns associated with the site are *learned* using methods analogous to anomaly detection systems or the early work of Henry Teng of DEC.

Aziz Mounji presented the ASAX Configuration Analysis System which checks for the presence of vulnerabilities in the configuration of a system, in the style of COPS and Kuang, but integrated with an intrusion detection system. This is the first attempt at an integration and thus would cause a check for a configuration error (such as a critical file becoming *setuid root*) as a result of a report from an intrusion detection system. In addition, the intrusion detection component could be triggered as a result of a report from the configuration checker.

Richard Lippmann is heading a Lincoln Laboratory project towards evaluating intrusion detection systems. Although currently sponsored by the Government, this technology could be applied to commercial intrusion detection products. The work has led to a testbed that includes data sources which, although synthesized resembles real data, were gathered off Air Force networks. To evaluate the testing methodology, it has been applied to the ASIM system (an early network sniffer which included statistical analyzers and signatures of attacks known at the time of CMAD). Richard noted that ASIM produces many false positives, which require the significant attention of human security administrators. Work is in progress to seed the test data with variants of known attacks and new attacks to provide a serious challenge for the intrusion detection systems.

Questions:
A lively discussion followed the presentations. The following questions were raised and discussed.

- **What is the impact of collecting and processing data?** For systems whose data source is network sniffers, the collection impact is minimal. Audit collection causes 3-4% performance degradation, although this low figure depends on collecting the "right data"; turning on full auditing can cause 15% performance degradation.

- **Is the reporting of suspicious activity flexible and understandable?** This has been a major feature of commercial systems.

- **Is the system easy to manage?** A concern is whether the system can be managed remotely.

- **Is the system scalable?** One issue is whether the capability of the system can be improved, for example, as new signatures become available. An interesting issue plaguing virus detection systems and real-time intrusion detection, is how systems

can cope with an increasing set of signatures. Clearly, preprocessing of the signatures can reduce the time to carry out pattern matching, but even this might not scale.

- **As new attacks are observed, can the signature database be updated automatically?** This would require abstracting the features of an attack that (1) can be observed from audit logs or network sniffing and (2) bear directly on the ability of the attack to achieve its goal. The participants were not optimistic about making this process automatic.

- **As new vulnerabilities are observed, can the signature database be updated automatically?** Many attacks, particularly against privileged programs or protocols, exploit vulnerabilities that are known to exist. Hence, intrusion detection systems would provide almost complete coverage against vulnerability-exploiting attacks if signatures could be generated automatically, given a specification of exploitable vulnerabilities. This was an area the participants recommended worthy of exploration.

- **Can the system detect previously unseen attacks?** Clearly, those systems that match data streams against patterns in a simple-minded way will fail to detect any attack that is not codified in the systems pattern database. Since it is very easy to construct an unbounded number of variants in almost every attack, this is a serious problem. State-based methods (such as D-STAT and Purdue's colored Petri net system) and the specification-based methods would work better. In addition, symptom-based methods would also succeed, such as systems that detect transitions to root not obtained by the acceptable commands.

- **Can the system detect multi-target attacks, distributed coordinated attacks and spreading attacks?** It is known that packet insertion or deletion can fool sniffer-based systems, either maliciously or as a result of natural network problems. In general, these kinds of attacks are not detectable by existing systems, as aggregation and correlation of activity from multiple sites are required, and the inference capability required is non-trivial. Some of the participants thought correlation is best considered in the context of large-scale intrusion detection systems, but the session on this topic explored this problem only briefly.

- **What is the false positive rate?** This turned out to be a controversial question. There is no denying that anomaly detection systems produce false positives, unless they are tuned to detect only behavior at the very fringe of the profiles they collect. It can be argued that signature-based systems produce no false positives since all activity they detect is, by definition, an attack. The question, however, is whether all detected "attacks" serious enough to warrant action, such as disconnecting the system from the network to initiate recovery. The answer is clearly no, when for example a "sweep" leads to a penetration. Some users report that false positive reports from existing systems is their major limitation. The solution seems to depend on improving the correlation and inferencing capability of the current systems, a research issue.

- **What is the false negative rate?** There is insufficient data to answer this question, although the experience of NADIR is promising. Clearly, signature-based systems will not catch new attacks.

- **Can the systems detect policy violations?** As long as all policy violations can be codified as attacks, it is possible. However, a policy is often better stated with respect to objects that abstract above the objects touched by an attack. Work is needed to produce a language that expresses a security policy in a manner that permits the translation of the policy to signatures that can be processed by an intrusion detection system. In current practice, an organization's security policy is expressed in natural language, if stated at all.

- **Is the intrusion detection system self-protecting?** The issue here is whether the intrusion detection system can be attacked. All intrusion detection systems seem to be vulnerable to Denial-of-service attacks (e.g., through flooding). The problem is exacerbated as network capacity increases. In general the problem of protecting an IDS seems to require the cooperation of intrusion detection components, perhaps based on the techniques of fault-tolerance. The problem is somewhat simplified by making it difficult for an attacker to break into an intrusion detection system, much as it is difficult to break into a router that is not intended for general purpose use. However, providing a remote management capability, gives a handle to attack an intrusion detection component.

- **Can intrusion detection techniques be applied to applications, such as the Java Virtual Machine (JVM)?** In principle, anomaly detection methods could generate profiles of activity by, for example, mobile programs making calls on the JVM. In addition, each site could create a policy statement indicating the calls on the JVM it finds acceptable. It is unclear if the JVM audits the activities that are appropriate to checking the policy or detecting attacks using anomaly detection, although the issue seems to be easy to decide.

- **Can intrusion detection be applied to database systems?** Participants discussed this issue in the session on misuse.

- **How are the features selected for anomaly detection systems?** Currently, they are selected manually. It has been suggested that anomaly detection systems should be specified as the problem of selecting an optimal classifier, where the determination of the minimal feature set is part of the classifier selection process. Search methods can be used, but if the candidate feature set is large, as in the application of intrusion detection to application programs, the search is very expensive. Heuristic methods would be useful, but there is little work to report.

- **What are the prospects for Windows-based intrusion detection systems?** Several organizations are working beyond UNIX targets, and as attacks on non-UNIX systems become known, it is likely that there will be a need for intrusion detection systems for other platforms. In principle, sniffer-based systems can be provided with

signatures to detect any attack, and many operating systems generate audit logs that could be the data source for host-based intrusion detection systems.

- **What are the pressing problems?** To summarize, the pressing problems include: policy-driven intrusion detection, detection of unknown attacks, correlation of reports from multiple intrusion detection components, protecting the intrusion detection system, detection of denial-of-service attacks, application of intrusion detection to application programs, and generation of signatures from a specification of vulnerabilities.

- **Does intrusion detection offer any scientific principles?** This question was raised in most of the sessions. It is the opinion of the editor that the principles are surfacing, but slowly. The reader is referred to the Introduction to these proceedings for details.

Session 3: New Attacks and New Twists on Existing Attacks

***Moderator and Session Editor**: Christopher Wee, Todd Heberlein*

Presenters: Christoph Schuba, Rob McMillan, Simson Garfinkel, Fred Cohen and Hai Ping Ko.

Speakers presented vulnerabilities and attacks in different domains including the WWW, cellular mobile phones, email and the Internet. The theme is that computer security is engaged in an arms race with malicious attackers. As existing vulnerabilities are closed, new vulnerabilities are exploited or old vulnerabilities are penetrated using more sophisticated attacks. New vulnerabilities are also continually introduced with the addition of new protocols, services and software into our systems (HTTP, WWW, Java, ActiveX).

Rob McMillan of CERT identified a downward trend in the number of reported security incidents. From 250 in 1995, the number of incidents in 1996 are decreasing. However, the incidence of high-impact incidents is stable or increasing. There are approximately 90 reports a quarter, which translates to 7 reports/week.

Type of attack	1st Qtr. 1996	2nd Qtr. 1996	3rd Qtr. 1996
Infrastructure	<1%	<1%	<1%
packet sniffers	5%	7%	7%
IP spoofing	2%	2%	1%
Root compromises	20%	20%	33%
Other successful	47%	44%	33%
Known unsuccessful	26%	27%	25%

Several speakers related "war stories" including denial-of-service (DOS) attacks against TCP/IP network stacks on hosts, vulnerabilities of the mobile cellular network, data driven attacks and perception management and DCA attacks.

The dominant form of new attacks are denial-of-service attacks, in particular SYN-flooding on IP networks. SYN-flooding exploits weaknesses in the implementation and design of the TCP/IP version 4 protocol and can prevent legitimate users from connecting to the victim host. Directed at the email, telnet or Web server ports, the attacker can deny those services to others. Many pieces of the Internet infrastructure (e.g., routers, gateways, firewalls) are vulnerable to SYN-flooding and similar attacks, so the attacker has tremendous leverage. With the increasing commercialization of the Internet, many sites are reporting high-value losses.

Denial-of-service attacks are not limited to network resources. Garfinkel showed how Web pages in HTML could allocate frames or windows until the operating system (especially Microsoft or MacOS) freezes or crashes. The nuisance attacks can exhaust our ability to respond to them; in essence a denial-of-service attack against our time.

Another popular type of denial-of-service attack is unsolicited email. Cohen related how an intruder could mount an indirect spam email attack by subscribing the victim to many mailing lists.

The common themes to denial-of-service attacks are:

- Resource exhaustion
- Sustained attack activity is necessary
- Attacks are anomalies with respect to resource consumption
- Timing is an essential element

The speakers and audience suggested various solutions to network DOS attacks including the adoption of "improved" TCP/IP version 6, delayed allocation of network resources, more tolerant implementations of the TCP/IP especially with regard to pending connections. It is possible to delay the allocation of pending connection slots in the TCP/IP stack by using state compression and cryptography to encode the TCP sequence number. Resources are allocated when the suitor is authenticated, and the connection is ready to carry data.

Another approach to network DOS attacks, suggested by Fred Cohen, is the adoption of stricter network security policies (e.g., Fred's zero-tolerance policy). By vigorously pursuing and responding to incoming network probes and attacks, the zero-tolerance stance seeks to discourage would-be attackers and also encourages intermediate parties to improve their security and not be used to "launder" attacks.

Accountability was one of the dominant security objectives desired. It is vital that we have a mechanism to trace network connections to their source. This capability is not presently available via the Internet infrastructure. Computer networks are approaching telephone networks in terms of performance, reliability and service requirements. Perhaps call-tracing which is available in telephone networks can be implemented for the Internet as well.

In summary, the panel made the following points:

- By attacking the Internet infrastructure (e.g., backbone routers or major software archive or WWW sites), they can inflict a lot of harm for a small amount of effort (i.e., leverage).

- Simson Garfinkel reported on denial-of-service and other attacks mounted through the popular downloadable mobile code (i.e., Java, ActiveX). Examples of such attacks are available at http://www.packet.com/garfinkel and Explorer Exploder, http://simson.vineyard.net/activex/exploder.html.

- Certificates are not a solution to the problem because the 1) use of certificates can be disabled, 2) the certificates can be forged, and 3) how users translate their trust of a

respected source into a signed certificate is debatable.

- Many attacks that occur are sporadic, so it is difficult to get vendors to respond. Because the average user wants speed and ease of use, vendors need to stop using C programming language. Some have switched to languages that are less susceptible to these problems. Corporations should sue vendors, and the federal government should take action against vendors as well.

- All solutions presented at this session will only work within the US. Information export is growing and will become a larger and larger industry. Economics will prohibit the fortressing of America.

- Big companies can get around export controls anyway, e.g. Exxon to China. A survey of 14 members about what companies need most revealed confidentially, accessibility and integrity. Availability is not even a requirement in the Orange book.

- Distributed Coordinated Attacks (DCAs) can disrupt email, mailing lists, and Web sites, usually by denying service. The probability of catching the perpetrator of a DCA is exponentially small, the more indiscretions he makes.

    - It is unlikely that we can prosecute attacks based on perception management. Perception management may be the only defense against itself. While zero-tolerance may work initially, it also is a two-edged sword and attackers can apply it against victim sites as well.

- NSA has a tool called parentage that can do correlation for audit records. Two papers have been published in Computers and Security. It is very complex to forge all audit trails.

- We need a taxonomy of attacks to clearly identify vulnerabilities that we haven't yet investigated.

In the future:

- The panel predicted increases in data driven attacks and email deluge. The malicious code in data-driven attacks is carried by data (e.g., documents or applets). As the Internet is employed as a direct marketing tool, spam email will increase, annoying users and wasting time and resources.

- Finally, the quality of security features in products and protocols is not increasing rapidly enough, thus we will expend even more time and energy to addressing intrusions.

Session 4: Intrusion Detection in the Large

Moderator and Session Editor: *Gene Spafford*

Presenters: Kevin Ziese, Stuart Staniford-Chen, Christoph Schuba, Doug Moran, Roy Maxion, Y. Frank Jou, Mark Crosbie, Joe Betser, and JF Mergen

The session, Intrusion Detection in the Large, was interpreted differently by participants—with topics ranging from large-scale attacks, large-scale systems, and a large scale of data, to a large locale being monitored.

One theme that was repeated in different contexts was that of scaling—being able to scale data collection, reporting, interpretation, monitoring, etc. The collection of systems we wish to monitor continue to enlarge, but the mechanisms in use do not continue to scale. This introduces delays and introduces capacity (storage, processing, bandwidth) problems. The problem is exacerbated by new sources of data as the systems get larger with additional routers, switches, and telecommunications infrastructure.

Several speakers suggested forms of aggregation as a means of addressing the scaling problem. Hierarchies were mentioned several times as one particular approach to aggregation of data collection, monitoring, and reporting. However, at least one person (Moran) argued that some systems are logically part of "clusters" that may overlap, and a strict hierarchy is not possible. In the end, the speakers did not agree on the best approach to such aggregation.

Several speakers related "war stories" about attacks or incidents they believed illustrated the problem, but only one (Mergen) discussed the problem from the standpoint of attacks on a multinational network. The discussions left it unclear if there was evidence of any credible attack that **required** a large-scale system. This suggests that some reduction of monitored systems may be a reasonable approach. That is, monitoring individual systems or groups of systems and aggregating reports may be sufficient rather than integrating data from all systems.

Several speakers also made the point that the people charged with monitoring the operation of systems may not have in-depth training in computers or telecommunications. Thus, the data presented must be trustworthy, clear, and direct. The user must know with certainty what is happening and how to react. Otherwise, if the output requires significant interpretation, it may not be used.

A few of the speakers seemed to imply that their research approach solved (or would solve) the overall problem, but the audience was unconvinced.

In summary, the following points were made by several of the speakers and panelists. Although these were not unanimous, they were largely accepted by speakers and audience alike when presented. They thus characterize the problem of Intrusion Detection in the Large as presented in the session:

1) There are many parties concerned with monitoring significant numbers of systems (including routers, switches, and telecommunications infrastructure as well as computers), often spread out over a large virtual space (potentially international).

2) Audit and monitoring data from all of these systems is difficult to collect, non-uniform in nature, and often untrustworthy as to content or timeliness. This is partly the result of a lack of standards and ad hoc collection methods, and partly a result of the distributed nature of the systems being monitored.
3) Audit data from attacks in large-scale systems is often incomplete and may contain contradictory information. (same reasons as #2)
4) "Typical" behavior for many large-scale systems may be difficult to characterize without eliminating useful detail.
5) Any data or alerts presented to humans monitoring large systems must be greatly simplified and condensed to allow for timely and appropriate response: time spent determining what several cryptic warnings mean may result in further loss or erroneous actions. The people using deployed systems may have little or no training in the computers they monitor.
6) There is difficulty understanding and representing the policies and practices necessary to drive an intrusion detection system, whether in the large or not. Until we can clearly express what we are watching for, it may not be possible to design a large IDS, or resolve the previous points.

Session 5: New Environments for Intrusion Detection

Moderator: *Marv Schaefer* ***Session Editor****: Karl Levitt*

Participants: Jim Anderson, Marv Schaefer, Deborah Frincke, Scott Cothrell, Katherine Price, Tom Haigh, Carolyn Turbyfill, Ab Kader

Although some of the early work used accounting records (IBM SMF) as the data source, intrusion detection has been centered on UNIX for the past decade. The reasons are clear:

1) Since many UNIX systems are open, most of the attacks have been directed against UNIX hosts or UNIX implementations of the protocol hierarchy.
2) The first attempts to produce C2 compliant auditing were for UNIX machines.
3) UNIX was not designed to be secure, so it offers many opportunities for attackers.

The situation is changing. More machines are now running Windows and NT than UNIX, and serious security problems are surfacing in the non-UNIX operating systems. There are many applications, particularly those that depend on the Web and distributed databases, that offer security problems independent of the hosts they run on. In fact, the entire issue of mobile programs is clearly a major security concern that is only beginning to receive the attention of the research community.

This session explored non-UNIX environments for intrusion detection. Since most of the work here is just starting, the presentations offered primarily problem statements and initial solutions. It is likely that there will be additional research in this area, and products that analyze NT audit records are beginning to surface.

Jim Anderson made a plea to the community about malicious programs contained in mobile data unbeknownst to its recipient. This is not entirely a new kind of attack. The "buffer overflow" attack inserts an arbitrary program into a stack (masquerading as data) that is executed when the return address is clobbered, so as to point to this program. Also, the macro-virus, associated with text editors and other applications, is in reality a malicious macro hidden in an otherwise innocuous text file.

Jim also proposed more complex and insidious attacks that, among other things, exploit the inconsistent ways that systems deal with fragmented packets. The kind of attack proposed consists of multiple stages on a system that is protected by a firewall. The initial part of the attack is to sneak a malformed packet that contains a small program through the firewall and avoid any auditing mechanisms. Initially this inserted program might install a sniffer that pokes around the system doing data mining, perhaps to inform the attacker what operating systems are being run on the hosts inside the firewall or scanning for security enhancements. Jim left the future progress of the attack up to our imaginations, but once the attacker is informed of weaknesses in the firewall and the hosts it purports to protect, there are many options for the attacker.

Picking up on Jim Anderson's theme, **Marv Schaefer** predicted major problems coping with data-driven attacks. Such attacks have been known from the early days of

computing. For example, the definition of Fortran does not prohibit the execution of arrays that could allow a program to inadvertently execute assembly language programs hidden in an array; Backus and Schwartz demonstrated this possibility. Considering problems with IBM System/370, Belady, et al., demonstrated that a program could execute "puns" instead of the code that was actually written, a security problem that would escape static analysis of the program being tricked.

Marv speculated on what the attacker could achieve by a data-driven attack, concluding that it is dependent on the privileges of the environment performing the execution. For a victim with privileges, there are few defenses: static analysis of data (searching for hidden programs) and audit. Although the former defense has been known and studied for many years, there are few practical techniques. It is likely that an attacker can defeat any static analysis program that is published. Audit, in principal, can check for critical objects being accessed. Marv also suggested that along with runtime detection, it is a good idea to consider automated recovery – a good idea in coping with any kind of attack or naturally occurring fault. This problem has been studied extensively by the fault-tolerance community, and has led to protocols that are now part of systems offering fault-tolerance. For example, in a transaction environment, the concept of serializable transactions accompanied by rollback/recovery is known and implemented.

Deborah Frincke suggested a paradigm shift: we should start considering object-oriented environments in our work on intrusion detection. Already, CORBA is being suggested as a system to use in the management and distribution of large data sets, especially over a distributed environment. If the intrusion detection community sticks with monitoring only operating system calls and using network monitors to detect known attacks against operating systems or the low levels of the protocol hierarchy, it will be useless in detecting attacks on applications. Deborah did not offer new solutions, but posed some provocative questions:

- **What should be the policy governing access to the data (through methods) associated with an object-oriented system?** Clearly, the policy can be dependent on the object and the site performing accesses.
- **Where should the policy be enforced?** Possibilities include the site maintaining the data, because it is responsible for the integrity and release of the data. However, a site receiving data might mistrust it, or suspect the presence of hidden malicious programs.
- **Where is monitoring performed, especially if the data is distributed?**
- **Where are audit logs maintained?** Perhaps the audit logs are just another object to be managed.
- **Can the objects be self-protecting and independent of where they are stored or accessed?**

Scott Cothrell made a passionate plea for the intrusion detection community to give attention to the Windows environments ('98 and NT), as the Government is purchasing more of these environments and downplaying Unix. On the contrary, most researchers eschew Windows operating systems, and it is only lately that commercial intrusion detection system vendors are delivering systems that detect non-Unix attacks. The key is

to determine what systems the research community should focus on. Known attacks will arise, and signatures can be developed as the attacks become known. Attacks on the protocol hierarchy will not be different from those considered for Unix. Denial-of-service attacks will surely be successfully launched. In the absence of public-domain source code, however, vulnerability-based defenses will probably not arise from the research community. Perhaps security through obscurity will keep the attackers at bay – for awhile.

Data-driven attacks, as discussed above, are likely to become popular, particularly in the light of paradigm shifts ongoing in connection with Windows systems. Similar to the research shift to active networks, there is interest in having systems be configurable on the fly as needs or the environment changes. For example, web pages will be created and modified continuously, allowing their remote management among other things. Additionally, in new distributed computing paradigms, programs and data of interest to a user can be anywhere on the Internet. Protocols will exist to permit the average and largely uninformed user to create a very dynamic and flexible distributed system. All of this poses major security problems for data that must be protected and new challenges for the intrusion detection community. Essentially, as posed by Deborah Frincke, it will be necessary for mobile data and programs to carry along their own security policy, and for sites to guarantee some form of protection or monitoring.

Anticipating the creation of CIDF (Common Intrusion Detection Framework), **Katherine Price** discussed the need for a standard format for audit data. Such a standard would promote portability and interoperability of intrusion detection products. In the current practice, each developer creates their own format, and it is impossible for tools to inter-operate. As new devices and applications that need auditing are created, the situation will become exacerbated. For example, there is a need to correlate data across platforms, (e.g., host operating systems, firewalls, routers), but it is difficult without a standard.

Katherine mentioned some existing proposals (ASAX, NADF, some work by Bishop on a standard for Unix auditing), but concluded that none are sufficiently general to cover the wide range of devices that are likely to produce audit data in the future.

The standard must include syntax and semantics – the latter necessary if the contents of audit data are to have an accepted meaning across platforms. Ideally, the format should be self-defining, perhaps driven by a specification of the device delivering the data.

Katherine also presented some thoughts on what data should be collected to facilitate their analysis by intrusion detection systems. For example, she presented the *binding problem,* where there is insufficient information in the audit data to resolve bindings in the presence of race condition attacks. Briefly, if a race condition attack occurs, the bindings in place before the attack may not apply after the attack. The file accessed is not apparent in the audit data, which returns the file name that could be associated with a file different than its binding prior to the attack. She also indicated that some redundancy in the information conveyed across calls could simplify the analysis carried out by an intrusion detection system. In general, this is a vastly understudied area and deserves consideration.

There will be tradeoffs between the need to keep audit data of manageable size but also useful for analysis. Redundancy is also desired to cope with attacks that selectively modify audit data.

Considering new domains for intrusion detection, **Tom Haigh** presented the Sidewinder Audit System. Briefly, the Sidewinder is an application-level gateway with proxies for standard services: generic TCP and UDP, encryption, etc. It relies on Type Enforcement to provide what is essentially mandatory security. Through type enforcement, it is possible to provide:

- System separations, where each subsystem (e.g., a sub-network) has its own set of domains and types that communicate through well-defined proxies.
- Lease privilege, which ensures each subject has access to only what it requires for what it needs
- Assured pipelines, so that the protection needs are carried across objects
- Role-based access control, whereby each user has its permissible domains in which it can operate.

Through type enforcement, a compromised application can cause only limited damage to other applications.

Of interest to CMAD, is the audit system in Sidewinder, which is actually a set of audit subsystems, each protected by type enforcement. Each audit system operates in its domain, filtering a specific kind of audit event and raising alarms specific to the system it is monitoring. There is the possibility for a response unique to the system being monitored, such as to send an appropriate email message to strike back to protect a service. In essence, each of the audit subsystems has the character of an *auditbot*.

Work is underway in evaluating the auditbot concept. A key issue in connection with a firewall (and any kind of protection mechanism) is its responsibility versus the monitoring system's responsibility. In the best of worlds, the firewall would block all traffic that threatens the system it is protecting. However, this is impossible because the firewall cannot be a significant choke point. Tom's approach, for now, is to use type enforcement to block traffic that violates coarse-grained access policies, and rely on intrusion detection for a more fine-grained analysis, such as to determine if a flooding attack or other more subtle attacks are underway.

Carolyn Turbyfill told the intrusion detection community that their concern for denial-of-service attacks might be misplaced. She acknowledges a fear that Java applets (and programs transmitted by ActiveX) can cause denial-of-service and other security problems. However, the task of administering a machine (network or even just a host) introduces its own denial-of-service problems, in part due to incompatibilities associated with newly installed software, and also the possibility of malicious code in this software. She poses scenarios where the porting of software is the source of massive denial-of-service problems that could span many sites, perhaps all simultaneously. Does the intrusion detection community have solutions to this kind of attack? Perhaps, a partial answer is to detect that service is degraded by new software, as the degradation might be

small and detectable through comparison with profiles. Recovery from the degradation is likely to be difficult, unless the source of the problem can be identified and rollback to previous configurations can be effected easily.

Ab Kader presented an overview of EPRI's security program, as the electric power utility companies move towards the use of the Internet as the media that links the individual companies. As the companies increase their cooperation through the Internet there is the specter of security problems – the kind that face all organizations. Moreover, there are initiatives to permit customers to communicate with the companies, which could allow additional avenues for misuse. For example, a program in San Diego gives customers the opportunity to access information on their power usage.

A security architecture is being developed, which includes penetration testing, real-time intrusion detection, and incident response handling. The CMAD participants were eager to participate in EPRI's research as the application is reasonably bounded (customers, for example, cannot run programs on the companies' machines), and there is a serious effort to create a security policy for the network that links the companies.

Session 6: Tools for Investigative Support

Moderator and Session Editor: *Becky Bace*

Participants: Kevin Ziese, Andrew Gross, Michael Neuman, Philip Reitinger

Introduction

Historically, as the long-time computer security research community worked on the problem of Misuse and Anomaly Detection, we dreamt of the day in which detection techniques would be effective enough to allow us to proceed to the next logical steps: investigating detected incidents, assembling the evidence pointing back to the guilty party, successfully prosecuting them, and then using the information gained therein to guide us as we proceeded to the next level of sophistication in detection methods.

Events of the past couple of years have brought us to the point where we've meandered through those post-detection events. The purpose of this session was to report insights gained in the investigative process and beyond to the community as a whole. We also covered the latest developments in the legal arena, which affect the rules limiting our activities in intrusion detection and response.

Questions Presented to Panel

1. What tools are required to do serious analysis of audit logs to determine the source of the attack?
2. What tools are required to do serious analysis of audit logs to do post-intrusion damage assessment?
3. What tools are required to do serious analysis of audit logs to do post-intrusion diagnosis of the weaknesses that allowed the attack?
4. What legal factors affect the use of incident diagnostic and forensic analysis tools in response to an attack?

The first panelist, **Capt. Kevin Ziese**, presented a compelling real-life example of a misuse detection problem in the U. S. Air Force Information Warfare Center. The problem occurred in the course of enforcing the United Nations restrictions on the activities of Iraq in the aftermath of the Persian Gulf War.

Simply stated, the task was to determine whether any use of Iraqi government computer systems was in violation of the U. N. restrictions (i.e. was there evidence of Iraqi computer use that supported weapons design and deployment or other prohibited activities).

Kevin presented an informative list of lessons learned in the course of addressing this high visibility, large-scale, real world application for misuse detection and computer forensic analysis technologies. Unforeseen complications he and his team encountered in the field included:

a. Comprehending files that were a language other than English, using non-Roman character sets,

b. Searching files in a language that ordered characters right-to-left

c. Searching Binary Applications Interfaces
d. Dealing with semantic data representation
e. Dealing with legal rules and restrictions, where there was not necessarily any correlation between the legal and moral concepts.

Cautionary notes and lessons learned include the following:
1. Solutions in the area of computer forensics need to be more scientific.
2. The goal must be solutions that fit problems, not problems to fit solutions.
3. Effectiveness must be prioritized before efficiency.
4. Tools must be designed in a less US-centric/OS-centric fashion
5. Security tools must be fielded with protective mechanisms such as serial numbers or encryption to allow installation and clean deinstallation on the fly.
6. In order to be considered usable, forensics tools must be designed so that they protect the privacy of system users.

The second panelist, **Andrew Gross**, is a Ph.D. candidate and senior researcher at the San Diego Supercomputing Center (SDSC). Andrew is the lead researcher for the Pacific Institute for Computer Security (PICS) project at SDSC, and has been researching tools for investigative support of incident handlers.

Andrew started his presentation with his "lessons learned" in the course of the last year's research:

1. "You can't do just one thing."
2. "Your tools can and will be used against you."

The remainder of Andrew's talk was devoted primarily to exploring the work he did in two functional areas of computer forensics, monitoring and encoding.

The work done in monitoring spans multiple networks (e.g. Ethernet, FDDI, ATM), and shows great promise. The tools were built based on the Berkeley Packet Filter software, and have many advanced features including dynamic monitoring with self-modifying collection filters. The detection rate, using even elementary filters coupled with the monitors, was impressive with over twelve intrusions detected and investigated in the first two months on-line.

The work done in decoding reflected both present and future considerations. The decoding features include dealing with Remote Procedure Calls, dealing with various network encryption features (Kerberos 5 and SSH), and replay capabilities in order to capture timing and other peripheral information about attacks.

Included in an extension of the monitoring tool research, tools were prototyped to analyze unknown binary programs, extract command information and other pieces of environment information from shell processes, and embed auditing features in kernel.

The third panelist, **Mike Neuman**, President/CEO of En Garde Systems, presented his view of doing interactive intrusion detection. He is using a sophisticated suite of network monitors and associated tools to monitor intruders, with the objective of determining

levels of expertise, objectives, and other factors that allow us to more effectively identify and prosecute system intruders.

Useful features developed by Michael and fielded by him in En Garde's software product line include tools to reduce packet traces to transactions, tools to allow one to look at network sessions from a variety of views (e.g. traffic levels only, transaction data, and raw traffic). He has also extended the intruder trap research started by Bellovin and Cheswick several years ago, by creating bait machines and using them to collect information about would-be system intruders.

Michael's view of needs in the CMAD area include the capability to create personalized intruder environments that spoof target systems environments for intruders. He also covered the strategy of randomly permuting the behavior of the targeted machine in order to either make the intruder uncomfortable enough to leave or else to keep the intruder connected long enough to trace back and apprehend. Michael also alluded to the legal questions associated with the whole issue of trace-back activities, which served as an excellent segue for our final presenter.

The fourth and final panelist for this session was **Philip Reitinger**, who is in the Computer Crime Section of the Department of Justice. Philip began with the issues brought up by Philip in the prior session, pointing out that hacking, regardless of who does it, is illegal, and that even when it is done in pursuit of a hacker, it can result in criminal prosecution.

Philip then presented information on the amended version of the Computer Fraud and Abuse Act (1030) which has now been extended to make more tracking procedures legal. In particular, he pointed out that some tracking is allowed in the case of Internet Service Providers who have been attacked, but that this still requires a Title 3 Order.

The fundamental point presented here is that certain sorts of computer tracking and monitoring have been defined as wiretaps, which are restricted by law. The point was also made that Computer Crime statutes are attempting to balance the need to protect information on-line with the right to privacy. There are many areas in intrusion detection, from audit data reduction to automated response to detected attacks, that are charged issues in legal circles. Philip concluded by fielding questions ranging from attacks on World Wide Web sites (the law has not decided whether WWW pages are a form of electronic communication) to the authorities to whom incidents should be reported (FBI).

Conclusion

The areas of investigation of incidents, computer forensics, and legal issues associated with misuse and anomaly detection remain of high interest to many. These are critical areas of interest to those that research and build CMAD systems, as they both constrain and enable critical functions in the CMAD process. Also, insight and eventual success in performing these functions are critical to containing loss resulting from intrusions and misuse, making the Herculean task of information systems security possible.

Session 7: New Ideas

Moderator and Session Editor: *Susan Gragg*

Participants: Mark Schneider, Peter Neumann, Gene Spafford, Mary Ellen Zurko, Bob Gleichauf, and JF Mergen

The intrusion detection field is very active and seems in some senses to be maturing. There is a lot of activity and research ongoing in the field and some products are now on the market and being widely distributed. If the field is defined narrowly as detecting intrusions from the outside, there has been enormous progress. However, if the field is defined as detection and prevention of attacks, no matter where they originate, then there are enormous challenges still ahead.

The goals of this session were to provide a place for new ideas in intrusion detection and to provide a "wrap up" session to discuss what we had heard over the entire conference. The two meshed together nicely and allowed the identification of some technical and other challenges that lie ahead for research and development in this field. This executive summary is divided into two sections – research challenges and development and/or productization challenges.

Research Challenges

In the area of research, there are at least four major challenges: **vocabulary development, anomaly detection, enterprise wide intrusion detection,** and **prevention/response.** Intrusion and anomaly detection needs a common language in order for everyone to discuss the issues, but there are many instances when a word such as "incident" is used in several different ways. What is the **vocabulary of intrusion detection**? Considering this and other recent conferences, it seems that people are beginning to understand what one is, but no one has written down a glossary of terms. That would be a useful research task. There is a great need for a common language.

Because new attacks may not fit known patterns, **anomaly detection** needs to emerge as a subfield. This means that we need the ability to distinguish between "normal" system operation and operation when a system is under attack – either internally or externally. Because systems are fragile and many system managers are not well trained, system problems frequently occur that could be poor software, configuration, or operation and maintenance. Frequently there is confusion between intrusions, denial-of-service, misuse and anomalies. How can someone monitoring the system tell the difference without looking after the fact at an audit trail? There are only a few people who can do this while there are many systems. There is a great need to understand program interactions, system interactions, user behavior and fault and exception consequences.

Generally, the field needs to focus more on **enterprise-wide intrusion and misuse detection.** Because the area is so UNIX centric, there are few tools and techniques to detect either intrusion or misuse in non-UNIX systems. The tools are available for those who monitor enterprise-wide systems are rarely constructed with integration with other tools or scalability to large systems, where large means at least tens of thousands of users. As other servers/systems are introduced, the community needs to expand it's thinking to

the development of more generic techniques that will work on non-UNIX and UNIX systems alike, provide analytic tools to less technical users and realize that techniques need to be scaleable.

Keep in mind that **prevention** is the main goal of any work in the intrusion detection area. During the conference, people discussed ways to partition off areas in systems for intruders to be observed. They also discussed potential **responses** to intrusions. This is an area that could be further explored and exploited.

However, if the existing products and ideas under development are to be successful, they have to go through two other stages – development and productization. Development is the stage where the work is turned into a product, whether the product is Government Off the Shelf (GOTS) or Commercial Off the Shelf (COTS). Production is introduced into a working environment and used to solve customer's problems. The most successful research and development efforts are based on an understanding of the customer's view of his problem.

Development/Production Challenges

In order for any of these research ideas to be widely used, they have to be available for "real" users as distinguished from "research" users. "Real" users do not always have degrees in computer science; some may have no degrees at all. The "real" users want to be able to tell the difference between intrusions, misuse and anomalies with a system that is easy to use, is infinitely scaleable, real-time, finds unknown conditions, is easy to maintain, updates itself and is completely portable across different systems. It is also important that there is some standardized testing of each intrusion detection system, so that it can be compared with other systems and users to determine which ones really fill their particular needs. The system also needs to be both initially affordable and maintainable.

Infrastructure support is another critical development issue that the community needs to confront. No one product or capability may be able to solve all the intrusion detection problems if they include anomaly detection and internal misuse. This means that the community needs to begin to think about an architecture for potential damage detection, prevention and recovery. The architecture may need to use several security tools together across broad groups of networks within an enterprise. Is that architecture hierarchical, distributed or something else? Someone needs to think about systems engineering of computer security systems, including some things that are usually avoided such as cost, maintenance and usability. This also means that someone needs to think about what an "ideal" computer security management scheme would look like and how it would mesh with legal and other enterprise-wide system management.

Another major research challenge concerns the fact that the work in the field is generally **U.S. centric**. Researchers should start thinking about the tremendous number of competent people in the rest of the world and realize that there may be other ways of doing things. They also need to recognize that these systems are increasingly tied to U.S. systems through mergers and partnerships and that they must be considered when looking

at both problems and solutions. Tools for forensic examinations of foreign computer systems may also be necessary when these systems are tied together.

This brings us to a final problem – the existing infrastructure. Organizations simply cannot afford to throw away legacy systems. The current infrastructure must be taken into account because there is so much of it. Adding security onto existing systems must not make the system slow down and must include ways to protect those legacy systems.

List of Participants

Anderson, Jim
James Anderson Co.
P.O. Box 42
Fort Washington, PA, 19034
215/646-4706
jpander@ix.netcom.com and/or
jpanderson@dockmaster.ncsc.mil

Aris, Brian
NSA
9800 Savage Road
Ft. Meade, MD, 20755
410/859-4537
blaris@missi.ncsc.mil

Attallah, Mikhail (Mike)
Purdue University
Department of Computer Science
West Lafayette, IN, 47907-1398
mja@cs.purdue.edu

Bace, Becky
Los Alamos National Laboratory
CIC Division, M/S B260
Los Alamos, NM, 87545
505/667-9904
bbace@lanl.gov

Berson, Tom
SRI International
764 Forest Ave.
Palo Alto, CA, 94301
berson@crvax.sri.com

Betser, Joe
The Aerospace Corporation
2350 E. El Segundo Blvd., M1-102
Los Angeles, CA, 90009-2957
310/336-0577
betser@aero.org

Bishop, Matt
UC Davis
Department of Computer Science
Davis, CA, 95616
916/752-8060
bishop@cs.ucdavis.edu

Brackney, Dick
NSA
9800 Savage Road
Ft. Meade, MD, 20755
301-688-0292
brackney@gnn.com

Bradley, Kirk
UC Davis
Department of Computer Science
Davis, CA, 95616
752-1287
bradley@cs.ucdavis.edu

Bradshaw, Gerald
NRaD
271 Catalina Blvd. Code 412
San Diego, CA, 92152

Champion, Terry
Rome Laboratory
RL/ERC-1
Rome, NY,
617/377-2068
tgc@rl.af.mil

Cheung, Steven
UC Davis
Department of Computer Science
Davis, CA, 95616-8562
916/752-1287-(lab)
cheung@cs.ucdavis.edu

Codespote, James
NSA
9800 Savage Road, Y441
Ft. Meade, MD, 20755-6000
410/859-6214
jpc@radium.ncsc.mil

Cohen, Fred
Sandia National Laboratories
Livermore, CA, 94550
510/294-4087
fbcohen@ca.sandia.gov

Collins, James
AFIWC
250 Hall Blvd., Suite 370
San Antonio, TX, 78243-7063
210/977-3134

Cothrell, Scott
NSA
9800 Savage Road, R232, Suite 6534
Ft. Meade, MD, 20755-6534
301/688-0847
sac@epoch.ncsc.mil

Crawford, Rick
UC Davis
Department of Computer Science
Davis, CA, 95616-8562
916/754-8380
crawford@cs.ucdavis.edu

Crosbie, Mark
Hewlett-Packard
19447 Pruneridge Ave. M/S 47-LA
Cupertino, CA, 95014
408/447-2308
mcrosbie@runner.cup.hp.com

Dacier, Marc
IBM Zurich Research Lab
Saeumerstrasse 4
CH - 8803 Rueschlikon, Switzerland
41-1-724-85-62
marc.dacier@zurich.ibm.com

Dilger, Mark
UC Davis
Department of Computer Science
Davis, CA, 95616
dilgerm@cs.ucdavis.edu

Essin, Dan
University of
Southern California
379 N. Encinetas Ave.
Los Angeles, CA, 91016
essin@hsc.usc.edu

Feiertag, Rich
Trusted Information Systems, Inc.
444 Castro St., Suite 800
Mountain View, CA, 94041
415/962-8885
feiertag@tis.com

Ferris, J. Martin (Marty)
Department of Treasury System Security
1500 Pennsylvania Ave., N.W., Rm 2415
Washington, D.C., 20220

Forrest, Stephanie
MIT AI Laboratory
545 Technology Square, Rm 814
Cambridge, MA, 02139
forrest@cs.unm.edu

Frank, Jeremy
UC Davis
Department of Computer Science
Davis, CA, 95616-8562
916/752-2149-(lab)
frank@cs.ucdavis.edu

Frincke, Deborah
University of Idaho
Department of Computer Science
Moscow, ID, 83846
208/885-6501
frincke@cs.uidaho.edu

Garcia, Roberto
Trident Data Systems
1100 NW Loop 410, Suite 607
San Antonio, TX, 78213-2257
210/442-4200
garcia@mailcenter.csap.af.mil

Garfinkel, Simson
Practical UNIX & Internet Security
P.O. Box 4188
Vineyard Haven, MA, 02568
508/696-7222
simsong@vineyard.net or
simsong@acm.org

Giordano, Joe
Rome Labs
315/330-8899
giordanoj@rl.af.mil

Gleichauf, Bob
Wheelgroup
13750 San Pedro, Suite 670
San Antonio, TX, 78232
800/494-3383
bobg@wheelgroup.com

Goss, Thomas
NSA
9800 Savage Road, R232, Suite 6534
Ft. Meade, MD, 20755-6534
tag@epoch.ncsc.mil

Gragg, Susan
Office of Research and Development
1820 N. Fort Meyer Drive
Arlington, VA, 22209
susang2@ucia.gov

Grance, Tim
NIST
820 W. Diamond St. Bldg 820, Rm 426
Gaithersburg, MD, 20899
301/975-3359
grance@nist.gov

Green, Kevin
Trident Data Systems
1100 NW Loop #410, Suite 607
San Antonio, TX, 78213
210/377-0477
email?

Gross, Andrew
UCSD Super Computing Center
P.O. Box 85608
San Diego, CA, 92186
6189/534-5086
drew@drew.sdsc.edu or
grossa@sdsc.edu

Haigh, Tom
Secure Computer Corporation
haigh@sctc.com

Halme, Larry
ARCA Systems
2540 North 1st St. Suite 301
San Jose, CA, 95131-1016
408/434-6633
halme@ca.arca.com

Hartman, Bret
Black Watch Technology, Inc.
2-212 CST/CASE Center, Syracuse U
Syracuse, NY, 13244-4100
hartman@blackwatch.com

Hashii, Brant
UC Davis
Department of Computer Science
Davis, CA, 95616
hashii@cs.ucdavis.edu

Heberlein, Todd
Net Squared
4324 Vista Way
Davis, CA, 95616
916/758-4338
heberlei@NetSQ.com

Heggestad, Hal
MIT Lincoln Laboratory
244 Wood Street
Lexington, MA, 02173
617/981-4014
heggestad@ll.mit.edu

Ho, Che-Lin
Cisco Systems
170 W. Tasman Drive
San Jose, CA, 95134
408/526-6840
che@cisco.com

Hoagland, James
UC Davis
Department of Computer Science
Davis, CA, 95616-8562
916/752-2149-(lab)
hoagland@cs.ucdavis.edu

Hofmeyr, Steve
University of New Mexico
Department of Computer Science
Albuquerque, NM, 87131
617/253-6625
steveah@cs.unm.edu

Irvine, Cynthia
Naval Postgraduate School
CS Department, Code CS/lc
Monterey, CA, 93943-5118
408/656-2461
irvine@cs.nps.navy.mil

Jackson, Kathleen
Los Alamos National Laboratories
MS B 265
Los Alamos, NM, 87545
505/667-5927
kaj@lanl.gov

Jennings, Toney
WheelGroup
13750 San Pedro, Suite 670
San Antonio, TX, 78232
800/494-3383
tjenn@wheelgroup.com

Jou, Frank
MCNC
3021 Cornwallis Rd.
RTP, NC, 27709
919/248-1409
jou@mcnc.org

Kader, Ab
EPRI - Electric Power Research Institute
3412 Hillview Ave.
Palo Alto, CA, 94303
415/855-2568
akader@epri.com

Karger, Paul
IBM TJ Watson Research Center
karger@watson.ibm.com

Kemmerer, Dick
UC Santa Barbara
Department of Computer Science
Santa Barbara, CA, 93106
805/893-4232
kemm@cs.ucsb.edu

Ko, Calvin
Trusted Information Systems
444 Castro Street, Ste 800
Mt. View, CA, 94041
415/962-8885
(x 3030)
ko@tis.com

Ko, Hai-Ping
GTE Laboratories
40 Sylvan Road
Waltham, MA, 02254
617/466-2510
hpk0@gte.com

Kosoresow, Andrew
University of New Mexico
Department of Computer Science FEC 329
Albuquerque, NM, 87131
505/277-8432
kos@mimbres.cs.unm.edu

Krsul, Ivan
Purdue University
Department of Computer Science
West Lafayette, IN, 47907-1398
317/494-9313
krsul@cs.purdue.edu

Kumar, Sandeep
Hewlett Packard
Cupertino, CA,
skumar@boink.cup.hp.com

Levitt, Karl
UC Davis
Department of Computer Science
Davis, CA, 95616-8562
916/752-7004
levitt@cs.ucdavis.edu

Lin, Meng-Jang
University of Texas at Austin
Dept. of Electrical & Computer Eng.
Austin, TX, 78712
512/475-6875
mj@pine.ece.utexas.edu

Lipman, Marc
Office of Naval Research (ONR)
lipman@onrhp.onr.navy.mil

Lippman, Richard
MIT Lincoln Laboratory
244 Wood Street
Lexington, MA, 02173
617/981-4014
rpl@sst.ll.mit.edu

Lipson, Howard
CERT Center, Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA, 15213
lipson@cert.org

Livingston, Paul
COSPO -still?
2N37 Plaza A
Washinton, D.C., 20505
703/281-8087
paulm.livingston-c-@da.gov

Mansur, Doug
Lawrence Livermore Nat'l Laboratories
mansur@llnl.gov

Marks, Don
NSA
9800 Savage Road
Ft. Meade, MD, 20755
301/688-0851
dgm@tycho.ncsc.mil

Marzullo, Keith
UC San Diego
Department of Computer Science and Eng.
La Jolla, CA, 92093-0114

Maxion, Roy
Carnegie Mellon University
Department of Computer Science
Pittsburgh, PA, 15213
412/268-7556
maxion@k.gp.cs.cmu.edu (or @K.GP.CS.CMU)

Mayfield, Terry
Institute for Defense Analysis
1801 No. Beauregard Street
Alexandria, VA, 2311-1772
tmayfield@ida.org

McFall, Steve
FBI Computer Crime Lab
202/324-9372

Meeson, Reginald
Institute for Defense Analyses
1801 N. Beauregard Street
Alexandria, VA, 22311-1772
703/845-6619
meeson@ida.org

Mergen, J.F.
BBN
9810 Patuxent Woods Dr.
Columbia, MD, 21046
jfmergen@bbn.com

Meushaw, Bob
NSA
9800 Savage Road
Ft. Meade, MD, 20755-6000
rvmeush@afterlife.ncsc.mil

Mitchell, Allison
Arrangements Coordinator
UC Davis, Dept. of CS
Davis, CA, 95616
mitchell@cs.ucdavis.edu

Moran, Doug
SRI International
AI Center
333 Ravenswood Ave. M/S EJ233
Menlo Park, CA, 94025
415/859-6486
moran@ai.sri.com

Moses, Melanie
NSA
9800 Savage Road, R22
Ft. Meade, MD, 20755
301/688-0292
memose1@alpha.ncsc.mil

Mounji, Abdelaziz (Aziz)
Institut d' Infromatique FUNDP
Rue Grandgagnage, 21 B-5000 Namur
BELGIUM, ,
+32-81-724987
amounji@info.fundp.ac.be

Myers, Eugene
NSA
9800 Savage Road
Ft. Meade, MD, 20755
301/688-0844
edm@tycho.ncsc.mil

Neuman, Mike
En Garde Systems
2101 White Cloud St., NE
Albuquerque, NM, 87112
mcn@engarde.com

Neumann, Peter
SRI International
333 Ravenswood Ave.
Menlo Park, CA, 94025
415/859-2375
neumann@csl.sri.com

O'Brien, David
UC Davis
Department of Computer Science
Davis, CA, 95616
obrien@cs.ucdavis.edu

Odneal, Sue
Kaiser Permanente
25 North Via Monte
Walnut Creek, CA, 94598-2599
510/926-3035
Sue.Odneal@ncal.kaiperm.org

Olszewski, Bob
Carnegie Mellon University
Department of Computer Science
Pittsburgh, PA, 15213
412/621-0933
bobski@cs.cmu.edu

Pace, James
UC Davis
Department of Computer Science
Davis, CA, 95616
pace@cs.ucdavis.edu

Palasek, Bob
Lawrence Livermore National Labortory
P.O. Box 808, L-303
Livermore, CA, 94550
palasek@llnl.gov

Peterson, John
NSA
9800 Savage Road
Ft. Meade, MD, 20755
301/688-0851
jep@tarius.ncsc.mil

Porras, Phillip
SRI International
333 Ravenswood Ave., EL231
Menlo Park, CA, 94025
415/859-3232
porras@csl.sri.com

Price, Katherine
Purdue University
Department of Computer Science
West Lafayette, IN, 47907-1398
kep@cs.purdue.edu

Puketza, Nick
UC Davis
Department of Computer Science
Davis, CA, 95616-8562
916/752-2149-(lab)
puketza@cs.ucdavis.edu

Reitinger, Phil
Department of Justice, Computer Crime Unit
1001 G St., N.W., Suite 200
Washington, D.C., 20001
202/514-4146 or
202/514-1026
woodenrabbit@os2bbs.com

Rosenthal, Rob
DARPA
3701 North Fairfax
Arlington, VA, 22203
703/696-2264
rmrosenthal@darpa.mil

Roy, Ray
NSA
9800 Savage Road
Ft. Meade, MD, 20755

Rozelle, Sharon
NSA
9800 Savage Road, R232, Suite 6534
Ft. Meade, MD, 20755-6534
301/688-0851
sar@tycho.ncsc.mil

Ruhl, Mary
NSA
9800 Savage Road
Ft. Meade, MD, 20755
301/688-0292
mkruhl@alpha.ncsc.mil

Saft, Mike
NSA
9800 Savage Road
Ft. Meade, MD, 20755
301/688-0293
mbsaft@alpha.ncsc.mil

Samorodin, Steven
UC Davis
Department of Computer Science
Davis, CA, 95616
916/758-9469
samorodi@cs.ucdavis.edu

Schaefer, Marv
ARCA Systems
10320 Little Patuxent Pkwy, Suite 1005
Columbia, MD, 21044-3312
410/715-0500
marv@md.arca.com

Schnackenberg, Dan
Boeing Defense and Space Group
206/773-8231
dan@baker.ds.boeing.com

Schneider, Mark
NSA
9800 Savage Road
Ft. Meade, MD, 20755
301/688-0851
mss@tycho.ncsc.mil

Schroeder, Mike
Digital Equipment Corporation
415/853-2215
mds@pa.dec.com

Schuba, Christoph
Purdue University
1398 Computer Science Building
West Lafayette, IN, 47907-1398
317/494-7814- or
317-746-5601-hm
schuba@cs.purdue.edu

Schultz, Gene
SRI Consulting
333 Ravenswood Ave.
Menlo Park, CA, 94025
415/859-5880
GSchultz@sibari.isl.sri.com

Sharps, Jennifer
Office of Research and Development
1820 N Fort Meyer Drive
Arlington, VA, 22209
703/613-8756
jennyks@ucia.gov

Shortliffe, Ted
Stanford University
Stanford, CA,
ehs@camis.stanford.edu

Shrobe, Howard (Howie)
DOD/ARPA
hshrobe@arpa.mil

Shutters, Chris
NSA
9800 Savage Road, R232, Suite 6534
Ft. Meade, MD, 20755-6534
wcs@tycho.ncsc.mil

Skelton, Ron
EPRI - Electric Power Research Institute
3412 Hillview Ave.
Palo Alto, CA, 94303
415/855-8753
rskelton@epri.com

Smaha, Steve
Haystack Labs
10713 RR 620 North, Suite 521
Austin, TX, 78726
512/918-3555x100
smaha@haystack.com

Smith, Brad
UC Santa Cruz
Department of Computer Science
Santa Cruz, CA, 95064
408/459-2370
brad@cis.ucsc.edu

Snapp, Steve
Haystack Labs, Inc.
10713 RR 620 North, Suite 521
Austin, TX, 78726
512/918-3555
snapp@haystack.com

Snow, Brian
NSA
9800 Savage Road
Ft. Meade, MD, 20755-6000
bsnow@dockmaster.ncsc.mil

Somayaji, Anil
University of New Mexico
Department of Computer Science
Albuquerque, NM, 87131
soma@cs.unm.edu

Spafford, Gene
Purdue University
Department of Computer Science
West Lafayette, IN, 47907-1398
317/494-7825- or
317-463-4857-hm
spaf@cs.purdue.edu

Stallings, Cathy
Los Alamos National Laboratory
MS B 265
Los Alamos, NM, 87545
505/667-5927
cxxs@lanl.gov

Staniford-Chen, Stuart
UC Davis
Department of Computer Science
Davis, CA, 95616-8562
916/752-2149-(lab)
stanifor@cs.ucdavis.edu

Steinberger, Ric
SRI Consulting
333 Ravenswood Ave. AH301
Menlo Park, CA, 94025
415/859-4300
ric@sri.com

Stelling, Paul
The Aerospace Corporation
2350 E. El Segundo Blvd.
Los Angeles, CA, 90009-2957
stelling@aero.org

Sterne, Dan
Trusted Information Systems
sterne@tis.com

Stolfo, Sal
Columbia University
Dept. of Computer Science, 606 CEPSR
New York, NY, 10027
212/939-7080
sal@cs.columbia.edu

Sullivan, Randy
NSA
9800 Savage Road
Ft. Meade, MD, 20755-6000
rls@tycho.ncsc.mil

Sundaram, Au (Robin) do
Purdue University
Department of Computer Science
West Lafayette, IN, 47907-1398
317/494-9313
sundaram@cs.purdue.edu

Sussman, Jeremy
UC San Diego
Department of Computer Science
La Jolla, CA, 92093
619/534-9669
jsussman@cs.ucsd.edu

Sutherfield, Lee
WheelGroup
13750 San Pedro, Suite 670
San Antonio, TX, 78232
800/494-3383
lsutt@wheelgroup.com

Swarup, Vipin
Mitre Corporation
202 Burlington Rd. M/S A128
Bedford, MA, 01730
617/271-2354
swarup@linus.mitre.org

Teal, Dan
WheelGroup
13750 San Pedro, Suite 670
San Antonio, TX, 78232
800/494-3383
teal@wheelgroup.com

Templeton, Steven
UC Davis
Department of Computer Science
Davis, CA, 95616
templets@cs.ucdavis.edu

Turbyfill, Carolyn
PGP
555 Twin Dolphin Drive, Suite 570
Redwood Shores, CA, 94065
415/654-3207
turby@pgp.com

Valdes, Al
SRI International
333 Ravenswood Ave.
Menlo Park, CA, 94025
415/859-4976
valdes@csl.sri.com

Van Wyk, Ken
SAIC
703/287-7685
ken@cip.saic.com or
krvw@mnsinc.com

Vu, Dai
Lockheed/Martin Communication Systems
M/S A&E-2E
Camden, NJ, 08102
609/338-4307
dvu@camden.lmco.com

Wade, Chuck
BBN
617/873-6260
cwade@bbn.com

Walnum, Scott
UC Davis
Department of Computer Science
Davis, CA, 95616
walnum@cs.ucdavis.edu

Wee, Chris
UC Davis
Department of Computer Science
Davis, CA, 95616-8562
916/752-2149-(lab)
wee@cs.ucdavis.edu

Wu, Shyhtsun Felix
North Carolina State University
Computer Science Department
Raleigh, NC, 27695-8206
wu@csc.ncsu.edu

Yemini, Yechiam
Columbia University
450 Computer Science Bldg.
New York, NY, 10027
212/939-7123
yy@cs.columbia.edu

Yip, Raymond
UC Davis
Department of Computer Science
Davis, CA, 95616-8562
916/752-2149-(lab)
yip@cs.ucdavis.edu

Zamboni, Diego
Purdue University
Department of Computer Science
West Lafayette, IN, 47907-1398
317/494-9313
zamboni@cs.purdue.edu

Zerkle, Dan
UC Davis
Department of Computer Science
Davis, CA, 95616-8562
916/752-1287
zerkle@cs.ucdavis.edu

Ziese, Kevin
AFIWC
250 Hall Blvd. Suite 370
San Antonio, TX, 78243-7063
210/977-3134
ziese@scorpion.cmet.af.mil

Zurko, Mary Ellen
Open Group Research Institute
11 Cambridge Center
Cambridge, MA, 02142
617/621-7231
zurko@osf.org

Survey Forms from the Participants at CMAD IV