**Special Report**
**CMU/SEI-93-SR-20**

# Results of a Workshop on Research in Incident Handling

**Thomas A. Longstaff**

**September 1993**

**Special Report**
**CMU/SEI-93-SR-20**

# Results of a Workshop on Research in Incident Handling

**Thomas A. Longstaff**

CERT Coordination Center



**Software Engineering Institute**
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This report was prepared for the

SEI Joint Program Office
HQ ESC/AXS
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER

(signature on file)

Thomas R. Miller, Lt Col, USAF
SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense.



This work was created in the performance of Federal Government Contract Number F19628-95-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 52.227-7013.

This document is available through Research Access, Inc., 800 Vinial Street, Pittsburgh, PA 15212.
Phone: 1-800-685-6510. FAX: (412) 321-2994. RAI also maintains a World Wide Web home page. The URL is http://www.rai.com

Copies of this document are available through the National Technical Information Service (NTIS). For information on ordering, please contact NTIS directly: National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161. Phone: (703) 487-4600.

This document is also available through the Defense Technical Information Center (DTIC). DTIC provides access to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential contractors, and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center, Attn: FDRA, Cameron Station, Alexandria, VA 22304-6145. Phone: (703) 274-7633.

# Table of Contents


**1 Introduction 1**
- 1.1 Background 1
- 1.2 Intent Of The Workshop 2
- 1.3 General Plan Of The Workshop 2
- 1.4 Participants Workshop 3
- 1.5 The Workshop Itself 4
- 1.6 Guide to the Proceedings 6

**2 System Development and Long-Term Issues Group Report 7**
- 2.1 Introduction 7
- 2.2 Discussion 7
- 2.3 Recommendations 9
- 2.4 Plenary Presentation 9

**3 Incident Handling Group Report 21**
- 3.1 Introduction 21
- 3.2 Discussion 21
- 3.3 Recommendations 23
- 3.4 Plenary Presentation 24

**4 System Integration Group Report 41**
- 4.1 Introduction 41
- 4.2 Discussion 41
- 4.3 Recommendations 43
- 4.4 Plenary Presentation 43

# Results of a Workshop on Research in Incident Handling

# 1 Introduction

## 1.1 Background

When the need for computer emergency response made itself known, it made itself known in a very public and theatrical way. One could say that at the time of the Internet worm, the Internet was in a crisis state, waiting for disaster to happen.

For many years, specialists talked about the potential for computer incidents of various sorts, but there was a respectable portion of the community that did not believe that the theoretical possibilities would ever emerge into practical concerns. Research into such topics received little attention.

As a result, when the imminent threat of computer and network incidents became widely known and the Computer Emergency Response Team (CERT[sm]) was established, the Internet was thrust into a situation where the decisions, developments, and inaction of the past had created a crisis situation in the present. The importance of the real-time emergency response mission made it necessary to focus almost exclusively on organizing for a continuous stream of computer emergencies, and to fight those fires first and foremost. Emergency situations had to take priority over proactive work that could preempt future problems.

Today, the CERT and other response teams face an ever increasing number of emergency incidents. The priority that is placed on fighting these “fires” remains very high. However, the rapid growth of the Internet, as well as the rapid advance of the supporting computer and networking technologies, make it impossible for incident handling to remain a purely reactive activity.

Without the investment of time and attention to address the growing problems of incident handling, to explore the relationships between incident handling and system design, and to widen the experience base of incident handling experts, our emergency response capability will erode to the point of being useless. Without a concerted effort towards devising proactive approaches for resolving crucial incident handling issues, developments in computer and networking technologies will continue to outpace the ability of the security community to devise countermeasures to the wide variety of computer abuse.

It was against a background of incident handling activities being dominated by a reactive approach to emergency incidents, that this workshop was held. The purpose of this workshop was to seek out fruitful areas for research and development that could expand the focus of incident handling, and narrow the gap between computer emergencies and the human and technological resources available to deal with them.

## 1.2 Intent Of The Workshop

The CERT Workshop on Research in Incident Handling was convened to address a wide spectrum of computer, network, and information security topics from the perspective of incident handling, in the present and future. The intent was to bring together researchers, incident handling specialists, users, system administrators, and managers to encourage an exchange of information and experience to mutual benefit. Specifically, it was to identify and roughly order, rich, full opportunity leverage areas for research and development, areas of particular interest to the Incident Handling function as well as areas that would benefit from the base of experience—and information—that the CERT Coordination Center has gathered, collected, and built up during its existence.

## 1.3 General Plan Of The Workshop

The workshop was organized into small-group working sessions focused on the intersection between system use, incident handling and research. A set of initial white papers on various topics of interest was provided to participants, before the workshop began. The plan was for the participants to review topics of interest, make suggestions for additions or deletions, and then organize themselves into smaller groups for closer examination of individual topics. Each topic was to be addressed directly, as well as assessed for importance and relevance to short-term, mid-term, and long-term time frames.

Discussions and conclusions of individual groups were reported in a brief plenary session at the end of the workshop. This report documents both the workshop itself and the preliminary conclusions and assessments of the material generated by the participants.

## 1.4 Participants Workshop

| Workshop Hosts | |
|---|---|
| name | affiliation |
| Larry Druffel | Director, Software Engineering Institute |
| D. Elliott Bell | D. Elliott Bell |
| Richard D. Pethia | Manager, SEI Service Workshops<br>Software Engineering Institute |
| L. Dain Gary | Manager of CERT Operations, CERT Coordination Center |
| Thomas A. Longstaff | Computer Security Researcher<br>CERT Coordination Center |
| D. Elliott Bell | Visiting Scientist<br>CERT Coordination Center |

| Workshop Participants | |
|---|---|
| name | affiliation |
| Dave Bailey | Galaxy Computer Services |
| Mark Barber | Defense Information Systems Agency |
| D. Elliott Bell | CERT Coordination Center |
| Matt Bishop | Dartmouth College |
| Marvin J. Christensen | Lawrence Livermore National Laboratory |
| Steve Crocker | Trusted Information Systems |
| Edward DeHart | CERT Coordination Center |
| Capt. Tim Grance | AFCSC/SREC |
| Cliff Hathaway | University of Arizona |
| L. Dain Gary | CERT Coordination Center |
| Jody E. Heaney | The MITRE Corporation |
| Michael V. Joyce | The MITRE Corporation |
| Lawrence J. Kilgallen | LJK Software |
| Thomas A. Longstaff | CERT Coordination Center |
| Robert McNeal | Defense Information Systems Corporation |
| Jim Molini | Computer Sciences Corporation |
| David Rosenthal | ORA Corporation |
| Steve Smaha | Haystack Labs, Inc |
| Theodore Ts'o | Massachusetts Institute of Technology |
| Mabry Tyson | SRI International |

| Workshop Facilitators | |
|---|---|
| name | affiliation |
| W. Carter Bullard | CERT Coordination Center |
| Mike DeRiso | Software Engineering Institute |
| James T. Ellis | CERT Coordination Center |
| Barbara Fraser | CERT Coordination Center |
| Norman E. Gibbs | Software Engineering Institute |
| Bill Hefley | CERT Coordination Center |
| Howard Lipson | CERT Coordination Center |
| Dennis B. Smith | Software Engineering Institute |

## 1.5 The Workshop Itself

The Workshop followed the general plan fairly closely. It began with a plenary session with warm welcomes from Larry Druffel (the Director of the Software Engineering Institute), Rich Pethia (the Manager of SEI Services of the Software Engineering Institute), and L. Dain Gary (the Manager of the CERT Operations of the CERT Coordination Center). The work began with a general consideration of the material entitled “Task and Time Frames. This material was distributed to participants for review, before their arrival in Pittsburgh, so a full-blown presentation of the topics was unnecessary. David Bell led the discussion of the material. The discussion then turned to the proposed issue of assigning participants into subgroups. The original proposal was to have four subgroups: (1) incident handling, short- to medium-term: (2) system integration, short- to medium-term. Based on the preferences of the participants, the group assignments were altered and the system-development and long-term issues groups were combined. The final subgroups were as follows:

| Incident Handling | |
|---|---|
| name | affiliation |
| Mark Barber | Defense Information System Agency |
| Matt Bishop | Dartmouth College |
| Marvin Christensen | Lawrence Livermore National Laboratory |
| Steve Crocker | Trusted Information Systems |
| Capt. Tim Grance | AFCSC/SREC |
| Cliff Hathaway | University of Arizona |
| L. Dain Gary | CERT Coordination Center |
| Bill Hefley | CERT Coordination Center |

| System Integration | |
|---|---|
| name | affiliation |
| Dave Bailey | Galaxy Computer Services |
| Michael V. Joyce | The MITRE Corporation |
| Lawrence J. Kilgallen | LJK Software |
| Steve Smaha | Haystack Labs, Inc |
| Theodore Ts'o | Massachusetts Institute of Technology |
| James T. Ellis | CERT Coordination Center |
| Barbara Fraser | CERT Coordination Center |

| System Development and Long-Term Issues | |
|---|---|
| name | affiliation |
| Edward DeHart | CERT Coordination Center |
| Jody E. Heaney | The MITRE Corporation |
| Jim Molini | Computer Sciences Corporation |
| David Rosenthal | ORA Corporation |
| Mabry Tyson | SRI International |
| W. Carter Bullard | CERT Coordination Center |
| Mike DeRiso | Software Engineering Institute |
| Howard Lipson | CERT Coordination Center |
| Dennis B. Smith | Software Engineering Institute |

The participants then divided into their subgroups and began deliberations. Over the period of Thursday afternoon and Friday morning, the groups debated on their various topics. The original plan to meet together as a whole group for status checks was quietly abandoned so as not to impede the flow of ideas. On Friday afternoon, the participants met as a whole for a final plenary. At that meeting, the chairs of the individual groups presented summaries of the topics discussed and conclusions reached, and also offered recommendations and suggestions for research directions and for future research workshops. That material is the subject of Sections 2-4 below. The Workshop was adjourned with thanks to the participants for their hard work.

## 1.6 Guide to the Proceedings

*Results of a Workshop on Research in Incident Handling* is organized as follows:

Section 1: Introduction

Section 2: System Development and Long-Term Issues Group Report

Section 3: Incident Handling Group Report

Section 4: System Integration Group Report

Sections 2, 3, and 4 are the results of the individual subgroups. Each section begins with an overview of the subgroup, specifically, the pre-identified topics and participants in the group. Next, there is a description of the discussions held by the subgroup; these subsections constitute executive summaries of the groups' results. Finally, a full record of the plenary presentation consisting of the groups' hand-out materials, with verbatim transcripts of the voice-over provided by the groups' chairs is given.

# 2 System Development and Long-Term Issues Group Report

## 2.1 Introduction

The System Development and Long-Term Issues group was formed by combining two groups, the System Development group, chaired by Jody Heaney, and the Long-Term Issues group, chaired by Jim Molini. The charter of the Systems Development group was to discuss identified and serendipitous system design issues as they relate to incident handling. The identified topics were as follows: the relation of system design to security engineering; the use of formal methods instead of or in addition to system design methods; the topic of software and vulnerability analysis in the system development process; and the idea of retrofitting, as it relates to system design. The charter of the long-term issues group was to consider the full spectrum of research topics that impact incident handling in the long-term. The white papers “Software Development and Security Engineering” by Jody Heaney and “Long-Term Issues in Incident Handling” by Jim Molini (distributed prior to the workshop) demonstrate the focus and point of view brought into the discussions by the co-chairs of this group.

This group included, at various times, the people listed below.

| name | affiliation |
|---|---|
| D. Elliott Bell | CERT Coordination Center |
| Edward DeHart | CERT Coordination Center |
| Jody E. Heaney | The MITRE Corporation |
| Thomas A. Longstaff | CERT Coordination Center |
| Jim Molini | Computer Sciences Corporation |
| David Rosenthal | ORA Corporation |
| Steve Smaha | Haystack Labs, Inc. |
| Mabry Tyson | SRI International |
| W. Carter Bullard | CERT Coordination Center |
| Mike DeRiso | Software Engineering Institute |
| Barbara Fraser | CERT Coordination Center |
| Howard Lipson | CERT Coordination Center |
| Dennis B. Smith | Software Engineering Institute |

## 2.2 Discussion

Four principal topic areas were discussed: the definitions related to incident handling; incident handling itself; incident prevention; and incident handling policy. The full plenary report that was presented to the entire workshop is included below, in Section 2.4. The description here summarizes the topics that are included there.

Within the area of incident definition, both the definition of “privacy” and “incident” were discussed. “Privacy,” it was felt, will change before ten years, possibly adding the variation that “privacy” could be location-specific. Specifically, the varying jurisdictions of privacy may continue to pursue different paths, resulting in privacy rights and expectations that depend on “where” along an internet working path one is interested. “Incident” is currently a hazy term, essentially a victim-defined crime. Improvements in the ability to ascertain and define “incidents” are laudable. Members of the computer security community are urged to aid in the legislative processes aimed at improving the preciseness of the term “incident.” Three technological factors were also noted as affecting the future definition of “incident” and “privacy”: the increasing use of encryption: the possibility of group or corporate authentication for various network accesses or services; and autonomous processes.

In the area of incident handling, four general topics were identified: audit-trail-related topics; the identity of the future’s incident trackers; the issue of physical location information; and the issue of identifying and analyzing vulnerabilities in regards to adequate auditing in preparation for incident handling reaction. Increased speeds and capacity will encourage screened audit recording and rapid recycling of audit-trail storage space, resulting in relatively short windows for identities exceptional information that needs to be kept longer. These shorter windows could make the recognition of some attacks of abuses, which are naturally longer than the circular queue or which purposely attempt to avoid detection by out-waiting the audit mechanisms more difficult. The question of whether there might not be a greater emphasis on host communications (in preference to network traffic monitoring) was raised for consideration.

Overall, it was felt that a number of effects will be seen, due to the anticipated increased volume of audit-trail data: improved means for data reduction, collection, and analysis; positive user identification to aid in screening and reduction; means for the combining of separately gathered audit data; and general understanding on the part of network users that part of the price of network (and host) access is being continuously subject to auditing and monitoring. The issue of who will be doing the tracking in the future was raised; classes of answers included enterprises themselves, service providers, CERT -like organizations, and law enforcement agencies. Furthermore, it was concluded that the means for determining the physical location of network users would be very important, the privacy impact notwithstanding. Finally, the issues involved in entrusting vulnerability information with outside organizations (for example, operation system vendors sharing the information with the CERT Coordination Center) will not disappear, but will rather intensify and become more complicated.

In the area of incident prevention, two topics were addressed: the importance and relevance of formal methods and place of software engineering. Regarding formal methods, there was a consensus that formal specifications have considerable value and disagreement between the applicability of more specialized tools and techniques. With regard to software engineering, it was concluded that security engineering should be built on solid software engineering. Some attention was given to the issue of how to bring pressure on product vendors to use bet-

ter software engineering practices. A final issue dealt with the fact that software reuse, particularly in the context of software reuse repositories, raises serious trust and confidence questions (such as “is this reusable code unchanged since it was deposited?”).

In the area of incident handling policy, there were three sets of question that were identified and a set of evolving technologies that could affect the incident handling policy. In the area of privacy, the group asked: “is intrusion detection an invasion of privacy?”; “what is a legitimate ‘expectation of privacy’ on the internet?”; Regarding who the incident handlers will be, it was concluded that CERT Coordination Center functions could be incorporated into a branch of law enforcement. It was further posed “in that case, what will the relationship of law enforcement be to the common carriers?” In the area of the class of internet users, it was asked how to deal with the paradox of having both the naive, privacy-expecting class of users and the more sophisticated abusers increasing at a very rapid rate. In the area of technology, it was asked whether encryption would solve the privacy problem and whether network monitors will become obsolete. It was also asserted that positive authentication techniques will affect incident handling in the future.

## 2.3 Recommendations

The System Design and Long-Term Issues group had four research suggestions:

- Research into audit trail standardization
- Research into intrusion
- Research into positive authentication techniques
- Research into combining security engineering and software engineering

It was suggested that future research workshops include the addition of a legal track.

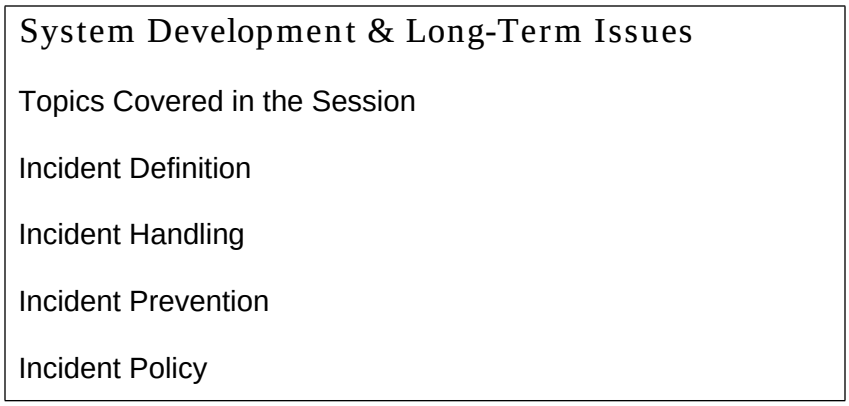
## 2.4 Plenary Presentation

This section records the presentation that the System Development and Long-Term Issue group made at the final plenary. The text consists of the viewgraphs used by the speakers, Jim Molini and Jody Heaney, interleaved with a nearly verbatim transcript of their voice-over.

Jim Molini Speaks:

We were handling long-term issues and system development issues, so you’ll see a system development orientation.

As you tell, we will be covering these four topics, quickly: incident definition, incident handling, incident prevention, and incident policy. We had very active discussions in these topics, and I’ll try to go through them. Have 20 minutes left, Tom? Okay.




**Slide 1**

Think of the system that you will see *after* the year 2000. That is the area in which we tried to focus on: we're thinking in terms of gigabit networks to PCs; we're thinking in terms of hundred-MIP workstations at your desk; we're thinking in terms of massively parallel processors and DBs; and many types of unusual and autonomous precesses floating around the network.

As a result, we expect privacy will probably be defined or redefined, depending upon your opinions, by 2000. We are fairly certain it will vary according to jurisdiction. And since you will be able to execute queries that may be global in nature, the definition of what privacy is and what your rights are going to vary, possibly even while your query is executing. These kinds of issues will have to be addressed by incident handlers and things like that.

What is an "incident"? We had real trouble defining what an incident is. We came to the conclusion that it's something like a victim-defined crime. It's some type of unauthorized—it should be "unauthorized" not "authorized." Sorry. "...access." Oh, well. Unauthorized access to systems or data and/or denial of service and that covers a broad range of types of things.

In order to define what an incident is, it's like saying "what is crime?" We need to participate in the legislative process in some way, shape or form. We have more recommendations regarding that later on.

We looked at evolving technology as far as incidents were concerned. We realized that encryption will affect the incident handling and privacy issues together. People will be able to have more privacy, but at the same time, people will probably have less privacy, in certain cases, as they use more global computer systems. We are not sure how pervasive encryption will really be at this point. It will be simpler; however, you will require either a common key which is generally known or it may require clear text transmissions. So it's very difficult to determine the level of encryption, and therefore, at what level we'll see privacy, and at what level we'll see the ability to do intrusion detection.

We'll see more transaction based processing as I described. And at points like when you spawn autonomous processes, it is quite likely to have to transmit your necessary authentication along with that process. Access control will be done by the individual hosts that you're going to be communicating with, but the authentication will have to be carried along with the process itself.

There are going to be differences. . . . There may be differences—we're not sure—as far as user versus corporate authentication. There may be many reasons why my company may not want everyone in the world to know which queries I make in a database across the country. They may not want everyone to know how I do my work, and when and where. As a result, to prevent things like traffic analysis, they're maybe going to say "You use our corporate id." I may not even know that the corporate system is making transactions on my behalf across the international network. If it is doing things on my behalf, the place where agreements are made, for example, who has access to NEXUS/LEXUS may be executed in the name of the corporation. There may be no need for me to have individual access. There is this differentiation and it could possibly lead to the need for some type of group level authentication.

The presence of autonomous processes—knowbots and things like that—will reduce the level of user control over what happens out here on these global networks. Users may execute processes, or start many processes, but not know what is happening "on their behalf." Can we really say "on their behalf"? It's a tough question. That was incident definition.

I have to apologize. Our 20-minute presentation can nowhere near indicate the lively discussions on each one of these issues, but maybe we can give you an idea.

## Incident Definition

Privacy

1. Will be (re) defined by 2000 evolve by society. Pushed by litigation and legislation.
2. Will vary according to jurisdiction

Incidents

We can't define it today; victim defined

1. Is it authorized access to systems or data, or denial of service?
2. Computer security community should be contributing to legislative process.

Evolving Technology

1. How will encryption affect incident handling and privacy? Consensus that encryption will affect IH, but not on how pervasive it will become
2. Transaction based processing + authentication. User versus corporate (group) authentication privacy issues both individual and organizational
3. Autonomous processes will reduce user control. Where does the liability reside?

**Slide 2**

When one we decide an incident might be occurring, we stated looking at what is an audit trail? What do you do to track an incident? to see what's taking place? Remember, a lot of these incidents may take place in a matter of hours, minutes, seconds, or milliseconds. If an incident can take place on those amounts of time, can we do real -time analysis? Or what kind of information can we store regarding those incidents when we may not know we've been had till some time after it has occurred. That means we have to capture and store information. If you can imagine a network of several thousand machines, where each of them or many of them may be connected to a 100 megabit per second token network. And each of those to a gigabit backbone. You can imagine the types of problems you're going to have with that amount of data. Can we capture and store all the network traffic? It's a real question.

We also determined that because there are some privacy issues involved too. It's quite likely that we may not have full information about what is inside those packets. So packet analysis may no longer be feasible i.e., What do you do if you can't do packet analysis at that point?

I'm not sure if you can read all this, but we went to 10 plus years out, and talked about how the increased volume of the data. We need improved methods for data reduction and collection. Yes, that is true, but we may only keep this stuff for a limited amount of time. We may keep it in a circular queue with only, say, 10-12 gigabytes worth of data, and have to recycle every few hours. It may cause us to look more favorably on things like positive user identification, and it is some definite means of identifying who this came from. This may provide assistance in that case because it may be easier to track what occurs. A small window of opportunity for evaluation what goes on may prevent certain detection techniques, may prevent us from seeing certain kinds of attacks. We're not sure. What we need is to have individual hosts' audit information combined by a post-audit organization. We need to have individual hosts decide what's right and what's wrong with themselves, and have that information sent back to a central monitoring system to analyze what is going on the network or organization entity. Is there a bandwidth problem with the transmission of all the data back and forth? Also, the users must sign up to the fact that we will be doing some form of rule-based analysis of monitoring of what they do.

Incident Handling

Audit Trails

- Can we capture, process and store all the network traffic?
- More emphasis on communication with host rather than monitoring network traffic
  - Ten plus years out, increased volume of data may require
  - improved means for data reduction and analysis
  - keep for limited period of time with requests for longer storage
  - positive user identification (e.g., at token level)
  - small window may prevent us from detecting certain events
  - need to combine with host audit info organizationally: can we request all this info from multiple hosts (bandwidth?)
  - user signs up—agrees to rule-based analysis

**Slide 3**

Who does the tracking? Who's going to do this incident tracking, when we get many large organizations connected by third-parties on extremely large global networks? We identified a term called "Communities of Interest", "Enterprises", "Organizations." This could be a university, it could be a group, anything. We have individual service providers that will connect these organizations, entities on larger networks. And there may be CERT-like coordination centers.

We'll talk about this later. And finally there are law enforcement organizations. We know the FBI wants a role in doing incident handling. These incidents are interstate traffic by nature. They may have more involved in tracking.

It becomes really important to know physical location information in order to do tracking. Since it is possible to hack the system while driving across Texas, you might need to keep track of location. You might need to know when I cross into New Mexico, in order to cut off my access to certain types of data. This is a really big privacy issue. If we have this kind of location information, we could tell that you are working from CMU today; and tomorrow from some [sensitive or embarrassing place]. You could be operating from some drug hostel. This is feasible if you include enough location information.

Vulnerability, identification, and analysis; Everybody knows you have to share vulnerability information with the organizations doing the monitoring. If we share that information, who do we trust? How and what do we share? We also decided that we can look forward to decades of arguing about how and whether you share vulnerability information.

I'm going to let Jody cover the other parts, being co-leader of this group.

### Incident Handling (Continued)

Who does tracking?

- Communities of Interest / Individual Enterprises
- Intermediate service providers (regional networks)
- CERT-like coordination centers
- Law enforcement (FBI wants a role)
  - physical location information will be critical for incident handling. Big privacy issue!
  - vulnerability ID and Analysis
- Trust issues for IH teams
- Will improve with public key technology
- We can look forward to decades of arguing about sharing vulnerability information

**Slide 4**

Jody Heaney speaks:

> *Third area that we felt was an important, large area was prevention. We also worked around with title. We had down a couple of different ones like "elimination" and "reduction," a couple of different things. We settled on "prevention." We're not sure which one would have worked out like we wanted. The big question that came up first was "Do Formal Methods have a role in incident handling?" Well, basically the consensus was that some aspects of formal methods (such as formal specifications) have considerable value across the board. There was a lot of disagreement about the terminology "formal specifications" or "correct specifications." But since we couldn't really define that, we decided we'd better go with formal specifications. Because that was something at least some people would understand. If you said to a formalist person "correct specification," they wouldn't have any idea what you were talking about. There was disagreement over the value in limited realms, but we didn't go through every piece and part of that.*

Another point is that we need to need to integrate with traditional software engineering methods. This is another area we feel fairly strongly about with most of us in the group. We need upgrades of the tools of formal methods. Also need integration with software engineering tools as well as the use of software engineering process tools and techniques. In terms of the software engineering tools as well as the use of software engineering process tools and techniques. In terms of the software engineering area itself, we did have consensus on the fact that security engineering will be ineffective unless it is built on a solid foundation of software engineering, which we don't see right now. We need more software engineering in the development of trusted products, or those of high-security products, of high-integrity products, of high availability products, whatever you want to call it.

There is sort of a strong belief that pressures on vendors from the various areas listed could be used to get good software engineering practices and procedures from the vendors. Those included marketing pressures, risk of litigation, evolving standards. Finally, there was this possibility of a future technique, that we surfaced, which was to have software product recall. Kind of an interesting notion. Might jar a few vendors.

The last area we talked about in terms of incident handling prevention were reused and repository issues. Which again is something of the future. As we move towards repositories, in particular, and we have trusted software in there, high-integrity software, high-assurance software, *safe* software. How is it going to be handled when it hits the repository? How do we know, when we take it out to reuse it, how do we get a warm fuzzy about that software that it's still as high quality as when it went in, as trusted, or whatever we're looking for? There are large understandability and usability issues related to that.

## Incident Prevention

Do formal methods have a roll in Incident handling?

- Some aspects of FM, such as formal specifications, have considerable value.
- There is disagreement over the value of more specialized tools and techniques.
- There is a need to integrate formal methods with more traditional software engineering methods.

Software Engineering

- Security engineering will be ineffective unless built on a solid foundation of software engineering.
- Pressures on vendors
  - marketing
  - risk of litigation
  - evolving standards
  - future software product recalls
- Reuse and Repository issues

**Slide 5**

The fourth major area that we covered was incident handling policy. This one came up because other topics kept pointing us to privacy. We saw that we had to discuss privacy policy. Basically, it came down to whether intrusion detection itself is an invasion of privacy. There is sort of an applied expectation of privacy a lot of people have when they're on the network. It may not be there, but they think it is present. We were wondering whether there will still be a general expectation of privacy in global network we get to in the future. One of the things, not truly a minority feeling, brought up and perhaps not agreed by all. This was the idea that users have to share the responsibility to ensure privacy in the network. That the user must provide some measures of his own if they expect privacy on the network. That again raises the issue of who does the actual incident handling.

That generated the idea that the CERT may be absorbed into a law enforcement agency. The reason for this is that no one but a law enforcement agency is going to be able to handle the legal issues associated with incident handling in the future. That raised another big question that we didn't really answer: what will the future relationship of law enforcement to the common carriers be?

Another portion that we felt would impact on policy on incident handling area was changes in technology. Basically, here we questioned whether encryption would solve the privacy problem. As Jim mentioned, we went back and forth on this. The reason is that there are some european countries where you can't send things over the net if they are encrypted because they won't allow it in the country. So we have questions on a global level about using it [encryption] that way.

That raised the issue of whether network monitors will become obsolete. This is not just data encryption, but other technologies as they come along. That in turn came around to the fact that crackers already have a lot of the encryption techniques. They have the public key approaches. Does that mean that in the long run they are not the useful in providing privacy?

Another area of technology that we noted as possibly having an impact on incident handling policy were positive authentication techniques. We felt those would definitely impact policy in the future.

Finally, there was an area that impacts incident handling policy that has to do with the changing nature of the user. We saw increases in two groups in the future, both the number of naive users and the number of sophisticated intruders. In terms of the naive users, there are those that don't even know they are vulnerable on the network. They don't really. . . . They're the ones that have an expectation of privacy, even though it's not really there. Of course, with the more sophisticated intruders, yes, as we develop new techniques to eliminate or reduce incidents on the network, at the same time the intruders are getting better and better technically, getting more sophisticated and more able to thwart our newer techniques.

These are the four major areas.

## Incident Handling Policy

Privacy

- Is intrusion detection in itself invasion of privacy?
- Expectation of privacy. Is there going to be a general expectation of privacy in the “global network?”
- User shares responsibility to ensure privacy

Who does incident handling?

- CERT may eventually be absorbed by a law enforcement agency
- Law enforcement/common carrier relationship
  - changes in technology
- Does encryption solve the privacy problem?
- Will network monitors become obsolete?
- Crackers are already into encryption techniques
- Positive authentication techniques will affect incident handling efforts
- Changing nature of user
- Increasing number of naive users
- Increasing number of sophisticated intruders

**Slide 6**

We had some research recommendations. One was audit trail standardization. This is standardization research in the area of audit trail. Basically, standardizing for formats for combining audit trails.

The second area was more research into intrusion detection. Such things as traffic analysis. There is reduction, compression. There is also pattern matching, using expert systems technology. A lot of those various kinds of things. We felt that there would have to be on-board host intrusion detection. A host will have to rely on itself for intrusion detection. Also an out-of-bound, . . ., out-of-BAND communications requirement.

Another area was more research into positive authentication techniques. This came up again and again and again. We need to be able to identify the individual and where they come from and all that kind of thing if we’re going to be able to do incident handling in the future.

Finally, further research in combining security and software engineering, pulling the two communities together. And trying to impact current standards efforts so that in the future we will be in a better position.

Research Recommendations

Audit Trail Standardization

More Research into Intrusion

- Traffic analysis
- On-board host detection
- Out-of-band communication requirement
  - more research into positive authentication techniques
  - further research into combining security engineering and software engineering

**Slide 7**

Now, we had proposals for future workshops. Well we didn't really *have* too much in the way of proposals for future workshops. What we did want to do was express sentiments and some thoughts on this particular workshop.

We did feel that it was a useful exchange for everyone who was involved. We felt that was important. We felt that the primary beneficiary would be the CERT. The upshot of that is, to skip down one here, we feel that the CERT needs to evaluate the results and benefits of this workshop and look at what comes of it to see if it makes sense to have another one or an annual one or whatever it is. We also felt that, or actually, we expect that the published results should be beneficial to its readers.

One thing that we did think. This was because we got so much into privacy issues. We felt it would have been useful, even at this workshop, to have some folks who have insight into the legal side. They could have helped us with insight into privacy concerns.

And that's about it from our group.

QUESTIONS/CLARIFICATIONS

TYSON: [something about responsibility, legal trade]

ANSWER: None heard.

BELL: Were hackers using encryption for their own uses? or undermining benefits of non-abusers?

ANSWER: Using for own purposes

Proposals for Future Workshops

Was a useful exchange of information

Primary beneficiary is CERT

We expect the resulting publication to be useful to its readers

We believe CERT needs to evaluate the results/benefits of the workshop

*Need a Legal Track at Next Workshop!*

**Slide 8**

# 3 Incident Handling Group Report

## 3.1 Introduction

The Incident Handling group was chaired by Matt Bishop. The charter of the Incident Handling group was to discuss identified and serendipitous incident handling issues in the short-term to the medium-term. The identified topics were as follows: communications during incident handling; audit focusing; incident handling procedures for intruder handling (including intruder detection, intruder monitoring, and intruder analysis); and malicious (including prevention, detection, and analysis). The white paper “Some Medium Term Issues in Incident Handling” by Matt Bishop (distributed prior to the workshop) demonstrates the focus and point of view brought into the discussions by the chair of this group.

This group included, at various times, the people listed below.

| name | affiliation |
|---|---|
| Mark Barber | Defense Information Systems Agency |
| D. Elliott Bell | CERT Coordination Center |
| Matt Bishop | Dartmouth College |
| Marvin Christensen | Lawrence Livermore National Laboratory |
| Steve Crocker | Trusted Information Systems |
| Capt. Tim Grance | AFCSC/SREC |
| Cliff Hathaway | University of Arizona |
| L. Dain Gary | CERT Coordination Center |
| Thomas A. Longstaff | CERT Coordination Center |
| Steve Smaha | Haystack Labs, Inc. |
| Bill Hefley | CERT Coordination Center |

## 3.2 Discussion

Six principal topic areas were discussed: communications, intruder topics, malicious code, audit, procedures, and usability issues. Within each area, individual topics were ranked and categorized by applicable time-scale. In some cases, the set of topics organized neatly as a research and development plan. In others, the topics were somewhat independent of each other. Recommendations are implicit in the rankings of the topics; the recommendations in Section 3.3 were derived from the highest-ranking topics within areas that were categorized as short or medium term. The full plenary report that was presented to the entire workshop is include below, in Section 3.4. The description here summarizes the topics that are included there.

In the area of communications, various electronic mail tools (such as filters and DSS), a “directory” (in case of an incident, who to call, how to call them, both safely and reliably), various policies, and the infrastructure topic of trusted dissemination were identified. Work on electronic mail tools and directories were viewed as the most lucrative.

The area of intruder topics was divided into intruder detection, intruder monitoring, and intruder activity analysis. The topics under intruder detection can be viewed as a development program for better intruder detectors. Highest in priority is misuse, a development program for better intruder detectors. The next highest priority is misuse detection models, building on definitions of “misuse” and “anomaly.” After that, research into determining the effectiveness of misuse or anomaly detectors in theory and testing detector implementations for fidelity to the theory. Last but not least, concerns characterizing patterns of actions on the parts of both users and the system and construction models of actions and attacks. Comparing actual experience with detectors to the models of actions and attack sequences provides feedback for updating the misuse detection models.

The second subarea in intruder monitoring. The first three topics concern parameters of intruder detectors: what to monitor (attributes? the level of monitoring?), protection of the system and the monitors from attack; and where the monitoring should be done. The fourth identified issue is the scalability of monitoring schemes: would a sheer size increase (from 1 host to 5 hosts or 1,000 hosts) make the scheme ineffective? The last topic is a tool for back tracking a suspected or known intruder. The first phase would be to develop the functionality of being able to trace a circuitous route from the intruder’s point of origin to one’s own system or network. The second phase would be to implement and field such a functionality into a trustworthy form.

The final subarea is intruder analysis. The highest priority would be achievable, modest analysis tools, based most likely on filters for various kinds of signatures. Next would be identity tracking and confirmation functionality. Then come data reduction tools and investigations into the granularity of protection of audit profiles and the simple analysis filters mentioned above.

In the area of malicious code, there were two topics related to prevention and six related to detection. The most important one was investigations into characteristics or signatures of malicious code. The second most important one (related to prevention) was precise criteria for being able to define “malicious code” that takes into account site-specific factors that can make the same activity malicious on one system and non-malicious on another. The third topic was tagging source code with the writer or writers for authentication and verification purposes. Next were security engineering and code analysis techniques applied to the problem of malicious code. Following them, a technique referred to as “boundary warnings,” where a trusted subsystem is added to the system to warn of anomalies (such as the FORTRAN compiler attempting the write to the .login file). A more difficult, longer-term problem of locking running processes and analyzing running processes for changes was listed seventh. And warranty issues, the other prevention topic, was listed last, largely because it is not a technical approach.

In the area of audit focusing, there were four topics identified. The first was standardization of audit data. The next was assessment of the impact of auditing or monitoring. Analysis of attack signatures would rely on having more carefully worked out and verified attack signatures available. A last, longer-term topic was the ability to analyze policy and usability characteristics. The issue goes two ways. Given a particular policy, what actions need to be audited in order to audit that policy effectively? And given a particular audit log, what actions and sequences of actions can be reconstructed?

In the area of procedures, trusted dissemination techniques, schemes for ranking system difficulties or weaknesses in hierarchies or in classes, and procedures for intrusion analysis and vulnerability reporting, were listed.

In the area of usability, a number of tools and techniques that are known technology and craft, but which are not available centrally for use, were gathered. Tools for incident handling, incident reporting, and tools for incident handling teams were identified. While this area was clearly not research, it was felt that it would be a valuable area to pursue.

## 3.3 Recommendations

The Incident Handling group had sixteen research suggestions:

- Research into electronic mail tools
- Research into an incident handling directory
- Research into models of misuse detection
- Research into misuse detector effectiveness
- Research into models of actions and attack sequences
- Research into parameters on intruder monitoring
- Research into back-tracking tools
- Research into modest intruder analysis tools
- Research into characteristics of signatures of malicious code
- Research into tagging source
- Research into boundary warning systems
- Research into locking and analyzing running processes
- Research into standardization of audit data
- Research into trusted dissemination techniques
- Development of usability tool kits

They had one suggestion for future research workshops, namely the inclusion of legally trained participants.

## 3.4 Plenary Presentation

This section records the presentation that the Incident Handling group made at the final plenary. The text consists of the viewgraphs used by Matt Bishop, the presenter, interleaved with a nearly verbatim transcript of his voice-over.

Matt Bishop speaks:

Title

The Incident Handling Short-to Medium-Term is a joint production of all these people. And they didn't put down David [Bell] and Tom [Longstaff], who came in and out several times and contributed quite a bit to the discussion.

Incident Handling Short- to Medium-Term

a joint production of. . .

| | |
|---|---|
| Mark Barber | Mark Bishop |
| Marvin Christensen | Steve Crocker |
| L. Dain Gary | Tim Grance |
| Cliff Hathaway | Bill Hefley |
| Robert McNeal | Steve Smaha |

**Slide 9**

**Topics**

What we covered was short- to medium- term and some of the research issues, we thought, turned out to skid into the long-term. "Short-term," meaning one to three years; "medium-term," meaning two to five, and "long-term," meaning after five, which is a little different than the time frame that was initially proposed. We talked mainly about communications, dealing with intruders, malicious code, auditing, procedures, and usability. Many of the discussions often cut across boundaries. For example, we started out doing communications, then went to audit. After about an hour of discussion of audit, we found we were talking more about intruder handling than about audit. So we went ahead and did the intruders, then malicious code and then came back to audit and so forth.

Topics Covered in This Session

short term: results in 1-3 years

medium term: results in 2-5 years

long term: results after 5 years

Major topics covers:

- communications
- intruder
- malicious code
- audit
- procedures
- usability

**Slide 10**

**Progress**

I'm following the format that was in the book with the slides. With regard to "Group Progress on Issues," we basically made some. What we focused on really was what were research topics. We tried to focus on that, but there was a lot of other discussion, which I'll try to bring up, as I talk about the research things.

Group Progress on Issues

We made some.

**Slide 11**

**Assignments**

Post-workshop Assignments

We didn't make any.

**Slide 12**

**Recommendations**

And research recommendations. Basically, we had recommendations in each of the areas that we considered.

Now, when I was talking to Tom before the workshop, he mentioned he was interested in a sort of “headline overview” of things, so blame this . . . what follows, on my misunderstanding or something like that.

Research Recommendations

done by area; see the following slides

**Slide 13**

**Communications**

First of all, in the field of communications. There’s the headline: “CERT” session Leader Speaks to Elvis.”

Communications

National intruder

**Slide 14**

CERT Session Leader Speaks to Elvis!

There were three. . . there were four things which came out of the discussion. The first one was the policies: who do you communicate with? How do CERTs communicate among themselves? What tools are necessary to do this? What sort of issues arise? That type of discussion.

Trusted distribution, of course, plays a role in this. It plays a role throughout the rest of this. And trusted distribution not only in the sense of how can you be sure that what you’re getting is what was sent in the first place; but also how can you be sure what you’re getting is really from a source that you can trust. In addition, how do you know. . . If you’re in the middle of

handling an incident, and you have to contact someone else, how do you know who to contact? That brought up the idea of the directory—you need to be able to figure out who to contact reliably.

This came up with the Morris worm, when one group called another group to find out what was going on. The second group was afraid that the first group was the attacker who had released the worm, trying to get information. *So* they basically terminated the conversation, and got back. . . and started communicating with the second group after they figured out who to call.

The other issue that's important is electronic mail tools. The worm essentially shut down electronic mail, because it overloaded the systems. You need somehow to be able to send electronic mail accurately, if it's available. This brought up the discussion of privacy enhanced mail. Which could also be used for trusted distribution in some sense.

A lot of the things we discussed too. Where we were trying to get a handle on what sort of models would work. We basically concluded that you do need models. There are a lot of models out there. A lot of this work is pulling things together. For example, electronic mail tools and the directory. A lot of the tools are out there, but it's not clear how they need to be integrated in order to provide something that would be useful and robust. And safe.

Communications

E-Mail tools (filters, DSS, ect.)

Directory - who, how, safe, reliable

Policies

Trusted Distribution

**Slide 15**

We then turned to the issues of how do you handle intruder detection, monitoring, analysis and resolution.

Intruder Detection, Monitoring, Analysis, and Resolution

2600

We Wuz Hacked!!!

**Slide 16**

What research issues do those bring? This originally, as I mentioned, started off as a discussion of auditing. And we got into the issue of audit trails for intrusion detection mechanisms, which brought about and shift. Now there were a number. . . we split it down into several subgroups, including detection. We started off with detection. The initial question was “what exactly is misuse?” How do you define it? We compared it to anomaly detection, that sort of thing. How do you characterize how effective this tool is? There are two issues here. The first one is when you go about building a misuse detector, you’re going to have to define which actions you consider “misuse.” How do you do that? What sort of model do you work from? How do you determine how effective that model is, what you’re catching and what you’re missing? The second one is how effective is the intrusion detection mechanism itself? How do you test these beasts? These were our second points. The third one, the third point was how do you characterize user actions and system actions? What are the measures? How do you define some sort of representation for attack sequences or sequences of commands used to break into a system in such a way that you can analyze records from the audit records. It turns out this is a little bit more subtle than you might think. You might think, well, Action A occurs, then Action B occurs, and then Action C occurs. There is a lot of ancillary information. Some attacks could depend very much on timing or race conditions. Some attacks may be launched by multiple processes. One user having four or five logins and attacking from combining two. . . launching an attack that requires actions from two streams. Multiple users may do it. How could you correlate all this information? Models of actions and attacks could presumably shed some light on this. All of this would feed right back into the misuse detection, because as you build your models and see how they work in practice, you’re going to get a better idea for what you can capture. Youcan go back up and look at the misuse detector, add those new patterns to it, and so forth.

In the discussion of whether these were short-, medium-, or long-term under the definitions that we talked about, the first two points seem to be short- to medium- term, because some of that work is already being done. The third one, with characterizing user actions and so forth, seems to us to be more medium- to long-term, because we’re only now beginning to get a handle on it. There are some representations out there currently, but it’s not clear how effective they are, other than they catch people. Building models of actions and attacks is a more long-term project.

In terms of which ones we thought, what the priority of each we thought, this is basically it. This is what we concluded after some discussion.

Intruder

Detection

- S-M Misuse detections (initial baseline could forever be enhanced-spy vs. spy)
- S-M Effectiveness of Misuse/Anomaly Detection
  - S-M Testing of Systems
- M-L Characterization of [user action, system]
  - End M-L Models of actions/attacks
- Update Misuse detection capabilities based on new user models

**Slide 17**

The next aspect of handling an intruder once you've detected it is monitoring. See what exactly is going on here. There again were a number of points that came up. The first one was how do you protect the intrusion detection mechanism from attack? The first thing an attacker does when he breaks into a system, is look and see: Oh, you're running an IDS system, let me give you my audit records, instead of the ones the system would really generate: or let me disable your auditing mechanism completely: I'm sure you'd appreciate the saved disk space. Things like that. Of better yet, if you can figure out where the monitoring. . . where the analysis mechanism is, just go over and turn it off. And that leads to system denial of service or attacking the monitors directly.

The next question that came up was, okay, what do we monitor? What is it important that. . . what do we want to analyze? What attributes do we use? How fine-grained are these? Do we want to monitor every single action? Or can we get away with monitoring some classes of actions, and not other classes? Also, where do we monitor? Do we only look at a host that we're analyzing? Or do we look at a network, or a LAN, or a larger area network or something like that? And then, scalability. If you're monitoring one machine, it's not too bad. How about if you have a thousand machines in your way? How does the problem scale? How can you handle it?

Finally the other issue which came up was more along the lines of when someone attacks a system, they usually just don't connect from their point of origin and attack your system directly. They'll tend to go from their point of origin to host A, then to host B, then to C, then maybe back to A and so forth. Bounce around the network a bit and then come into your system. The question that was raised was how hard would it be to track from the system under attack to the point of origin? It turns out that this is an extremely non-trivial problem, for two reasons. The first is [that] you're relying on intermediate mechanisms, or on intermediate hosts over

which you don't have control. So the attacker could come in, completely. . . . *If* you had something equivalent to the *ident* protocol, the attacker could come in and completely spoof the *ident* protocol and then continue on through other hosts. So even if you did try an automatic trace back, you would get incorrect information. The second thing is that it's possible the system administrator on the other systems might not want to give unrestricted access to their systems . . . or unrestricted access to the relevant information of their systems to do the tracing. The common way to do this sort of thing is to run a program, at least in the UNIX world, *netstat*. That tells you where people are connected to you from. You can then run *finger* on . . . as and IP address. You then run *finger* on the system as an IP address and you work backwards that way. However, to run *netstat*, you need a login. System administrators might object first of all to giving logins like that out to anybody. Secondly, they might also object to the extra information that *netstat* reveals. They might not want that information getting out. This problem turned out to be probably the most, well, one of the hardest ones to deal with. Building a tool to do this, assuming this system administrator is willing to install it and is willing to cooperate, is very straight forward. It's basically just making *netstat* look like *finger*. The hard part is building something that can actually be trusted. *If* you build something like *finger*, it can be trusted in the same sense that the ident protocol is trusted: you know this is what the machine on the other end is telling you. Whether or not it's true, you have no idea. It's simply what the other machine is telling you.

Again this is pretty much how we classify things. For protection from attack, the interaction with existing systems we felt was medium-term, simply because there are mechanisms out there that can be used, assuming you trust your computing base. If you don't trust you computing base, it immediately becomes long-term.

Intruder Monitoring

M-L What to monitor-attributes level of monitoring

M-L Protection from attack system denial service/monitors

- Integration with existing systems hardening.

L Where to monitor

L Scalability

- Backtracking through multiple connections

S New Tool

M-L Open and trusted tracing facilities

**Slide 18**

Once you've got the intruder in there and you've got your data about what the intruder was doing, how do you analyze it? There are a number of issues, some of which Jim and Jody touched upon. First of all, you've got this humongous mass of data, maybe three or four edge-feet. How do you pick out the entries that are relevant, how do you reduce the data to a manageable form? What do you reduce and what do you discard? In particular, you have to be careful of not discarding something that is part of an attack signature, part of a description of the attack that you're interested in. They may be very, very obscure. So you have to be very careful at the point.

The second thing that we talked about was trying to find some goals for analysis. In other words, given that we have this mass of data, and we want to reduce it, we'll need some sort of a filter to look for the proper signature. That is much shorter term, because if you know how to describe the attack and you have a sufficiently powerful language, you can simply use that to do the reduction.

The next question was the granularity of the audit profiles and protection of the audit profiles. How do you protect these things? And how fine grained should they be? Some of which we talked about before.

And finally, the notion of identity tracking and confirmation. The problem here is not so much that when someone's on your system, who are they? It's more, when someone comes in two different ways to your system, how do you correlate the two? It's not so much figuring out who the individual is. More along the lines of what processes is that individual using?

We very deliberately in the group tried to stay away from deciding social or legal issues, or dealing with them, simply because with the legal issues, we didn't know enough. We agreed fully with both Jody and Jim Threat, if this is held again, there should be legal people involved.

## Intruder Analysis

S-M Achievable, modest analysis goals—filters for attack signatures user definable signatures

M-L Identity tracking and Confirmation

M Data Reduction

- What to reduce
- What to discard
- Signatures of attacks

M Granularity/Protection of these Audit profiles/filters

**Slide 19**

From intrusion analysis, we figured that people often, intruders often like to do nice little things like leave you little presents, you know like a doctored bin/login file. So we went to talk about malicious code from there. This is computer viruses, Trojan horses, and things like that. By the way, that was a real headline. Not in the New England of Medicine. I have it posted up on my door.

## Malicious Code Prevention, Detection, Analysis, and Resolution

*New England Journal of Medicine*

**“Man Infected by Computer Virus”**

**Slide 20**

The first issue with malicious code is how you prevent it. And immediately we discovered that we couldn’t really define malicious code. Something that is malicious on one system may be considered perfectly proper on another. So you have to have to come up with some sort of a definition that essentially ties into the security policy. And you need documented criteria for that. The other issue that was brought up was, why not sue the contractor? or sue the person who wrote it? That brought up warranty issues. And the priority we assigned to it was way on the bottom. The reason we did that was that from the technologic point of view, it doesn’t really help to know that you’re going to be able to sue someone: you still have to deal with malicious logic. From the legal point of view, most of these other issues are probably irrelevant. So we took a technologist’s point of view about priorities.

We then talked about taking and analyzing it. One of the first things we talked about was analyzing a running process. If you can infect a process that is executing, or after it as it runs, you can do quite a bit of damage. Would it be possible to lock the process so you can't alter it? The answer is "sure," as long as your operating system is safe. If your operating system gets hit, forget it. Also that you need operating system support for that. Then we talked about code analysis. There are many different types of code analysis, some of it not really useful, and a lot of it quite useful. Some of the things we talked about here were analyzing, for example, a pattern of writes that was not appropriate to the program. For example, if your FORTRAN compiler suddenly started altering the .login file or a .bat file, that would probably not be a very good sign. There are currently systems which are prototypes, and are being tested and worked on, which will do some things like that. They will analyze patterns of writing. You can also analyze system calls, patterns of system calls and things like that. So we thought that it was short-term to look at the prototypes; medium-term to try to improve them. And more importantly, short- to medium-term in figuring out what to look for.

In security engineering, we agreed completely that it depends very largely on software engineering. And that is an area that does need to be explored more.

Another thing that came up was identification of sources. If . . . it's true that warranties are mostly in the legal realm, but it's also true that if you can tag source with a specific individual or group that is responsible for that source, and then can make sure . . . have them somehow sign it, then if it's transmitted and it turns out . . . it's received correctly and it turns out there's a Trojan Horse or virus in that, that group will be quite embarrassed, so there is some deterrent effect there. So we spent some time talking about how to tie the identity of the source of a program to that program itself. The paradigm I used was the USENET. It can be handled with techniques very similar to privacy-enhanced electronic mail.

Detection of characteristics and signatures. This is rather like your PC virus directors, which look for specific things and will report problems. It can be anywhere from short-term to long-term, depending on how big you want to make the problem. Of course, it could be infinite too, given the undecidability problems.

The final one was boundary warnings. This is essentially based on a scheme of Karger's, which said that you build a trusted subsystem which sits between the operation system and application and when ever the application goes to access a file, the trusted subsystem steps in and checks a rule-base to see whether or not that is allowed. And if it is allowed, it goes ahead and does the access. If it's not allowed, it either terminates the program or asks the user, should this be allowed? Part of reason that we labeled it "short-term": is there was a scheme like that which was implemented on Unix by Nick Bly and Terrence Gray, which did something like that. Unfortunately, they made certain assumptions that vitiated most of the usefulness of the program. This is probably medium-to-long-term. The assumptions they made were that the shell, editor, and several other things were trustworthy. [Laughter]. That's usually where the attackers go first. For those who haven't been involved in that particular thing. However, the long-term issue, of course, is trying to come up with an automated way of

determining which files a program will access and how. In the medium-term, one could do something which would be fairly simple and fairly short. There are also variants on this scheme. So it is somewhat worth looking into.

Malicious Code

Prevention

2. Definition of malicious code—documented criteria

8. Warranty Issues

Detection / Analysis

7. M Locking / Analyzing running processes

5. S-M Code Analysis

4. M-L Security Engineering

3. S-M Identification of Sources (Authentication, Verification)

1. S-L Dietician characteristics / signatures

6. S-M-L Boundary warnings

**Slide 21**

Given all that, we went back to audit, because we figured by looking at things new we figured we had a better handle on what audit records would be appropriate, or what sort of audit information we would need.

Audit Focusing

Washington Post

GAO Report “Hough to Odit Sistm Repurts for Speling Erors” Finds No Problems

**Slide 22**

And again we got back to attack signatures almost immediately. We then started talking about some more theoretical notions of auditing, such as the policy and usability. In other words, if I have security policy X, I need to look for violations of that policy. From that policy, can I figure

out exactly what I want to audit on a real system? And another question was, given a log, how precisely can I determine exactly what happened to cause certain events? For example, if a file suddenly has changed protection modes, how can I figure out what happened? What sort of information do I need to keep? And what makes it interesting is . . . . An ancillary question is, given a log, can I reconstruct every event on the system? And the answer is, in general, no. So, what are the limits of that?

Another question was analyzing audit profiles. How do we do that? That was covered in an earlier slide. Standardization of audit data: We labeled that as a fairly high priority because we do want some sort of standard, or multiple standards. Everybody here knows what Tanenbaum said: standards are wonderful; that must be why there are so many of them. We figure that this way we could combine records and look at different types of trails created by different machines and try somehow to start to unify them. We think that would be a positive influence.

And then the next question about auditing is what is the impact of running an audit, or running audit mechanisms? How much does it hurt your system? What is the cost/benefit ratio? There are business effects. For example, you'll have to have someone go through the records or you'll have to build some tools to look through the records and so forth. So we considered all that. Most of these were short- to medium-term. There are some long-term issues in there, but most of what needs to be done would be more on the medium-term.

Audit Focusing

3. S-M Analysis of attack signatures

4. S-M Policy / Visibility

- Given Policy X, what do I need to look for, analyze, etc.?
- Given Log Y, what can I determine happened?

On Going Audit Profiles (see earlier detection/analysis)

1. S-M Standardization of Audit Data

2. S-M Impact Assessment (Incident/Monitoring)

(Cost/Benefit, business case (Time, $, People, Perception)

**Slide 23**

From that on to procedures incident handling.

Procedures For Incident Handling

Revolting Systems Administration Bulletin

How to handle intruders:

- Locate them
- “Let them Eat Cake”

—M. Antoinette

**Slide 24**

Again trusted distribution techniques reared its head. Clearly if there is a bug fix or a security hole, you want to make sure that it gets out to the people who you consider the (quote) good guys and good gals (unquote). We also had a lot of discussion on what constitutes the good guys or gals. The problem is that there is considerable ambiguity of role. You may have someone who is a good responsible system administrator by day, but by night turns into the Hacker with a Mohawk and will break into systems. So how do you reconcile this? We took the easier approach: it’s a social issue, we’ll worry about that later.

The question was how do you classify vulnerabilities or incidents? The issue here is that you’re generally not usually willing to announce to world “hey, we’ve discovered . . . someone has reported this serious flaw in the SUN operating system.” But it might be when someone asks you does this operating system have any serious flaw . . . . What is a “serious flaw?” By some sort of classification scheme, you could . . .might be able to determine how to handle a particular incident or vulnerability. Should you put it on the front burner or on the back burner? Should you push to determine how to handle a particular incident or vulnerability? Should you push very hard? or How? This also brings up how you might classify vulnerabilities into groups, to try to extract information that will let you predict or give you ideas of where to look for other vulnerabilities. All of that could fall under procedures and not much is known about that. We thought it was well worth looking into

Another short- to long-term issue is procedures for intrusion analysis and vulnerability reports. And to a degree that crosses with hierarchy and classification schemes. Another issue there, of course, is if I think something is very sensitive and I want to pass it to another CERT unit or another CERT organization, can I label the data? And can I have an agreement so that they will agree not to distribute it widely? The answer is, probably . . . . Well, we don’t know. I don’t think there are any such agreements in force.

Procedures for Incident Handling

1. S-M Trusted Distribution Techniques
2. S-M Hierarchy / Classification Schemes
   - S-L Procedure for • Intrusion Analysis
   - Vulnerability Reports

**Slide 25**

And then the next question . . . the last part of the thing that we looked at was usability—policies, procedures, and techniques. There weren't really research issues here, but there was a wealth of things that could be done and needed to be done.

Usability—Policies, Procedures, Techniques

Installation Procedures for Today

1. Open box
2. Locate installation procedures
3. Install on system

**Slide 26**

We talked about incident handling tools. A help desk organizer so that if an organization is setting up some help system so someone can call and say "somebody has just attacked my new system; what do I do?" You could get help that way. Or if a security patch came out, and you need to know how to install it, you could call up and get help that way.

Investigative tools. There are a lot of tools out there that will help you track down problems or help you track done problems or help you trace people who intrude onto your system. You can make this into something more formal: build a tool kit, so to speak. I think something like that is under discussion.

Protecting tools and facilities. How can you be sure that the tool that was posted somewhere, I don't know. Say, xyzzy.ftp.anon.com, or whatever, and anonymous FTP facility, hasn't been altered. There should be some mechanism for that. And also what Steve Smaha calls the "Hacker Whacker Catalog," the catalog of tools. Keeping track of all the tools that are out there. Not necessarily distribute or assured by or vouched for in any way by the CERT, just "Where are they?" So if someone wants to try them, they can go out and get them.

The next issue is incident reporting. When someone reports an incident, how do you deal with that? There were a number of issues. The first one was a subscription. If someone would be able to report information and get information back. Basically, there would be a need to be some sort . . ., it was felt that you would need some sort of agreement [in]. You would at least need to pay for a certificate to use some sort of privacy-enhanced mail or some sort of mechanisms to allow the information and vulnerabilities to flow safely. Binary inspection tools to see whether or not you had in fact the latest release of the system. Or to give information about the system.

We also would need a testbed for vulnerabilities. What brought this up was the comment that you can test . . . . If a bug report comes in for SUN— Sun is this glaring security hole—you try it out. Yes, it's there. Then you report it to the SUN. You may issue some sort of oblique announcement. Or at the end, when SUN has resolved the problem, you'll release a bug patch. The problem is that this vulnerability may exist on other systems. You don't know until you've tried it. There would need to be a testbed so when a vulnerability report comes in, you could try it on a number of systems, not just the one on which the problem was reported. A lot of these vendors use systems with common roots. Often it is true that what happens on one system can be done on a number of systems. So setting up this testbed . . . . it's not so much a research topic, but something that needs to be done.

For the incident handling team, we talked about trusted dissemination. How do you disseminate in a trusted manner? What do you give out? Do you give out raw data, IP addresses, user names, and such? That might be embarrassing, and is completely unnecessary. How do you protect what you're disseminating?

And then to help an incident handling team keep track of incidents, categorize them, classify them and that sort of thing. Bug tracking tools. Electronic mail filters to try to winnow out some of the chaff. DBMS to keep track of vulnerabilities or incident reports and then some mechanism for filtering dissemination.

Usability—Policy, Procedures, Tools

Incident Handling Tools (Site / Domain Level)

- Help Desk Organizers
- Investigative Tools
- Protection of Tools/Facilities
- Catalog of Tools (“Hacker Whacker” Catalog)

Incident Reporting

- Subscription (Pay for Certificate, Dissemination)
- Binary Inspection Tools
- Experiments / Testbed for Vulnerabilities

IHT - Incident Handling Team

Trusted Dissemination - How, What, How to Protect?

Tools

- Bug
- E-mail Filters
- DBMS
- Filters Dissemination

**Slide 27**

**Final Things: Suggestions [no slide]**

The final thing, which I don’t have a slide for, time got a little tight there at the end, was proposals for future workshops. We didn’t really talk too much about this. One thing that came out loud and clear was again what Jody and Jim had said. It would be nice to have legal people when this is held again. In particular, I’m not sure . . . . This is my opinion, I didn’t talk to other people, but I feel that it should be legal people scattered throughout the other groups. That was it.

# 4 System Integration Group Report

## 4.1 Introduction

The System Integration group was chaired by Dave Bailey. The charter of the System Integration group was to discuss identified and serendipitous system integration issues that relate to incident handling. The identified topics were as follows: integration issues that arise on a single machine, either by the consumer or by the vendor; integration issues that arise in a multi-machine context, such as interfaces, unitary login, and coordinated audit; the problem of distributed authority within the InterNet; and retrofitting products and systems and the security implications thereof.

The white paper “Issues for System Integration with Regard to Incident Handling” by Emily Lonsford (distributed prior to the workshop) demonstrates the focus and point of view brought into the discussions by the intended chair of this group. (Dave Bailey graciously agreed to chair this group when Emily Lonsford was suddenly unable to attend the workshop.)

This group included, at various times, those listed below.

| name | affiliation |
|---|---|
| Dave Bailey | Galaxy Computer Services |
| D. Elliott Bell | CERT Coordination Center |
| Michael V. Joyce | The MITRE Corporation |
| Lawrence J. Kilgallen | LJK Software |
| Theodore Ts’o | Massachusetts Institute of Technology |
| James T. Ellis | CERT Coordination Center |
| Barbara Fraser | CERT Coordination Center |

## 4.2 Discussion

Nine principal topic areas were discussed: user interface; security administrator interface; administrative issues; audit; problems relating to the cost of security; policy; vendor-related issues; standardization; and miscellaneous topics. The full plenary report that was presented to the entire workshop is included below, in Section 4.4. The description here summarizes the topics that are included there.

As initial topics, the group discussed both time frames and the scale of integration in order to focus their attention. They concluded that they would concentrate on medium-scale integration (integration that requires some coordination and some advanced planning; on a rough scale of two to five years from the present). Examples of what would be considered medium-scale were a typical air force base, a single laboratory or department at a typical university, or a national laboratory computing facility.

In the area of user interface, the principal difficulties were the absence of consistent access control interfaces and the absence of good access control interfaces. An additional item was user insistence on avoiding intrusive security measures and mechanisms. The potential solutions were a common security model, the promulgations higher-level interfaces, and standardization of user commands.

In the area of administrator interface, the same type of difficulties were noted: the absence of consistent security administrator interfaces and the absence of good security administrator interfaces. The potential solutions included some form of canonical management protocol and general encouragement of the use of consistent interfaces to security tools.

In the area of administrative issues, there were three noted difficulties. The first difficulty for a security administrator to know with assurance what all the effects of an administrative action will be. The second related to the necessity that computing resources follow changes in roles outside the system. For example, when a staff member is re-assigned from Accounts Payable to Accounts Receivable (a well-defined notion in the business would), a security administrator needs to be able to match those changes (both additions and deletions of compatibility) within the context of the company's computer network. The last has to do with the development of standard perspectives and tools for dealing. This role changes (as in RFC 1244) on the one hand, and tools for the configuring and maintaining system security, on the other.

In the area of audit, a long list of difficulties was assembled: the need for a tamper-proof audit trail; the need for secure audit trail transport; canonical format and content for audit trails; the need for support for application-level auditing; the need for readily available intrusion detectors; the need for generally available time synchronization tools; and attention to balance between privacy concerns and the needs of auditing. The potential solutions fell into topics for focused research workshops (compiling a list of needed audit data from various sources; development of a canonical audit trail format; and the development of a standardized real-time audit-trail transport mechanism) and the suggestions that existing technologies, such as WORM drives, be made use of more broadly.

In the area of problems that arise because of cost considerations, most of the problems stemmed from the existence of legacy systems (systems you wish would go away but stay put out of anthropomorphized spite) and the fact that users resist intrusive security with great vigor. All of the potential solutions were variations on candidate courses of action for behavior modification and were not really topics that could be profitably pursued by computer security researchers.

In the area of policy, there were three main difficulties noted. Two related to privacy: privacy is often not considered within enterprise of internet policy considerations, and the expectation of privacy by the larger community is not really known. The third difficulty had to do with the expression and comparison of policies for different products: policies are hard to state, and even when stated, they are hard to compare. The potential solutions were research to study the policy expression problem and attention of privacy and the expectation of privacy in the statement of enterprise policies.

In the area of vendor issues, there were a number of difficulties noted, all revolving around things vendors do that we'd rather they not do: pursue their own competitive advantage around proprietary products (such as RSA); disregard of the computer security community's desire for products with better security characteristics; false advertising; and delivering systems with security-feature defaults of OFF. All the potential solutions were hortatory and non-technical.

In the area of standardization, the topics of encryption, assignment of TCP/IP ports, and non-helpful Department of Defense guidelines on networks were noted. No solutions were identified.

Listed in the miscellaneous set of topics, the need for secure unitary login; the difficulty of mutual authentication of distributed hosts; and the paradox of wanting to shield assessment details from the general public while wanting to distribute assessment results widely. No solutions were identified.

## 4.3 Recommendations

The System Integration group had eight research suggestions:

- Development of a common security model
- Research into a canonical management protocol
- Research into standardizing role changes
- Research into tools for implementing role handling policies
- Research into tools for configuring and maintaining system security
- Research into policy statement and comparison
- Research into security unitary login
- Research into mutual authentication of distributed hosts

They had three suggestion for future research workshops:

- Constructing a list of needed audit data from various sources on the net
- Development of a canonical audit trail format for Incident Handling purposes
- Development of a standardized real-time audit trail transport mechanism

## 4.4 Plenary Presentation

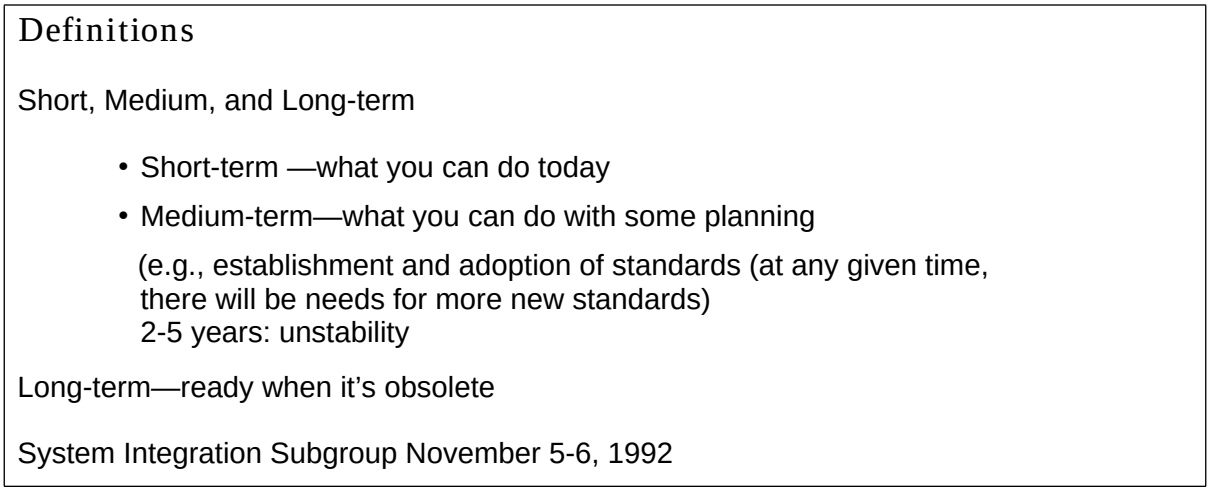
This section records the presentation that the System Integration group made at the final plenary. The text consists of the viewgraphs used by Dave Bailey, the presenter, interleaved with a nearly verbatim transcript of his voice-over.

Dave Bailey speaks.

Definitions

Well, since we're running a little late and everything I wanted to say has already been said, I'll simply ask if there are any questions. [laughter, applause] No, hunh?

We, probably like everyone else, began by talking about the time scale. We had several characterizations of the time scale. Feeling that ranges of years probably weren't quite appropriate, we did several different kinds of things. What we gave you here is probably the most facetious of them. Short-term things are what you could do today and get some benefit out of. Medium-term things are things that you could do fairly soon, but with a little advanced planning being required. And long-term things are things that probably won't be ready until they're obsolete, at least in some sense. But in other terms, we're talking about middle-term being 2 to 5 years, in terms of getting useful things out.




**Slide 28**

**Scale of Integration Issues**

Since we were the System Integration group, we also talked about the scale on integration. That is of interest, feeling that the issues for small scale integration were quite a bit different from those for large-scale. There are quite a few different factors that are important, things

such as whether there is common authority over the whole network, the numbers and sizes of things, the degree of homogeneity versus heterogeneity, and to some extent the availability of tools that will give you global information.

Scale of Integration Issues

Factors defining scale

- authority
- numbers / size of []
- heterogeneity
- availability of tools

**Slide 29**

**Definition of Medium**

So we characterized, tried to characterize, medium-scale integration by finding things that would distinguish it from large- and small-scale. We decided for example, one way to characterize small-scale integration was things you just could go and do without necessarily having to consult with somebody else. Medium-scale integration at least requires coordination with other people, and probably some advanced planning. Then we distinguished medium- from large-scale by the issue of common authority. That is, anything that doesn't have common authority over an entire network, for example, we said was large-scale. Here we meant not just "is there an authority?", but "would it really be effective?" For example, the President has common authority over the entire U.S. Government, but the idea of posing integration policy questions to the President just doesn't seem workable. We agreed that large-scale integration issues were important, but we also agreed to work only on medium-scale issues, since those seemed to be the technological issues. And the larger-scale issues involved more political, social, and legal issues. We thought that those were important issues, but didn't deal with them and I think we would suggest that's a subject for a future workshop.

Definition of Medium Scale

Coordination needs

- Small: just do it; grokkable by one
- Medium: ned to coordinate with others

Have line authority, common boss that is effective
(distinguishes from large but not small)

Agreement to focus on medium scale

System Integration Subgroup November 5-6, 1992

**Slide 30**

**Examples of Medium Networks**

We put down a few examples of medium-scale kinds of things. The networks at the national laboratories all fall in the medium-scale. A typical air force base might also . . . - I see an ear perking up there - might also fall in the medium-scale range. A large campus system where there's a great deal of autonomy between departments may actually fall in the large-scale areas, even though the number of nodes might not be greater than some of these at the networks.

One of the things I wanted to point out was . . . you see, Jim Molini mentioned discussing networks of several thousand nodes. By our definition here, that's probably a medium-scale issue, not a large-scale issue.

What we did then was to write down a large number of potential problems and try to cluster them into several issue areas and then came up with potential solutions, or partial solutions, for the issues. What I'm going to do is very quickly show you the areas and the kinds of problems that we came up with. I'm not going to make any claim of completeness anywhere. And I'm very carefully going to go quickly so you won't be able to force me into mapping solution to specific problems.

Examples of medium scale networks

- Los Alamos
- Air Force base
- MIT - LCS group (but not entire MIT campus)

System Integration Subgroup November 5-6, 1992

**Slide 31**

**User Interface Problems**

I'm going to go through these, sort of, in a sense, from things that could be done, or at least understood how to be done, fairly easily down to things that we kept pushing to the end of the list because we didn't see how to deal with them. User interface problems, for example. There's a whole cluster of issues here. Lack of consistent access control interface across several systems is a source of problems. Lack of a *good* one, consistent or not, is also a source of problems. And then there's the social problem that when you get right down to it, users don't want security if it's intrusive. They don't want not to be bothered by security people any more than they want to be bothered by intruders. There is a serious problem of people wanting security, but not being willing to pay the price for it.

User Interface Problems

- Lack of a consistent access control interface for users
- Lack of a good access control interface for users
- Users don't want intrusive security: e.g. tokens, frequent password changes, machine generated passwords

System Integration Subgroup November 5-6, 1992

**Slide 32**

**Possible User Interface Solutions**

Possible interface solutions. These are all . . . One of the things we did too was to try to characterize these as short-, medium-, or long-term issues. That didn't get on this viewgraph. You'll have to try to get the tapes from us to find out what we thought about that. Privacy problem. [laughter] [Question from audience: Do those tapes have an 18-minute gap? more laughter] There are several 18-minute gaps, in fact.

Let's see; I think I won't go through these specifically. Except that I did want to point out that we observed . . . . We talked about user interface issues and administrator interface issues. We thought there was a great deal of similarity here and we thought we ought to be seeing solution ideas that look similar, and yet they looked quite a bit different. That struck us as being a problem.

Possible User Solutions

- Common Security Model (research)
- Higher-level interface
- Standardization of user commands (add common commands)

Note: We are concerned that these solutions appear so different from those for the administration interface—seems as though they should be similar.

System Integration Subgroup November 5-6, 1992

**Slide 33**

**Administrator Interface Problems**

The problems with the administrative interface look a lot like the problems of the user interface: lack of consistent one, lack of a good one, across systems, . . .

Administrator Interface Problems

- Lack of a consistent security administrator interface
- Lack of a good interface for security administrators

System Integration subgroup November 5-6, 1992

**Slide 34**

**Possible Administrative Interface Solutions**

And yet things we thought of here for doing something about that seemed to be quite a bit different than the things we thought about for dealing with user interface issues . . . You can read those.

Possible Administration Interface Solutions

- Canonical Management Protocol - Standards-based solution
- Nurturing software / hardware vendor competition
- Like SNMP, but for security management
- Ask the IETF to look into this (again, similar to the way SNMP was adopted)
- Encourage use of consistent interfaces to security tools

System Integration Subgroup November 5-6, 1992

**Slide 35**

**Administrative Issue Problems**

There are a lot of other administrative issues. These sort of center around the administrator of a network being able to understand and keep track of the state of the network. The awareness of what the real effects of administrative changes are going to be is a hard problem. It's not always obvious that the changes you make administering a network are confined locally. They may have much broader implications that you can see.

Handling role changes referred to changes in role for the users, and this is a context issue more than technological issue in the network. It's the idea of somebody moving from Account Payable to Accounts Receivable, for example, and having to change the system to correspond to that. We also thought in terms of system security controls: consistent is good; good is good; simple would be nice, too. Our idea of "simple" was a slider on a window. Like to be able to set my complete security condition by moving a slider up and down. So at two-thirds, that's too hard to use; so I move it down to a half. This is perhaps a long-term issue.

And then, Continued Assurance of System Security. Here we wanted . . . we were looking for something like the current security state, which is available not only to administrators but to users. You might want to know the current security state of a particular system before you sign on to it. It's like: "are you there? who else is here? are you secure?"

Administrative Issue Problems

- System administrator awareness of administrative actions
- Handling role changes
- Simple security configuration controls
- Continued assurance of system security

System Integration Subgroup November 5-6, 1992

**Slide 36**

**Possible Administrative Issue Problems**

We then identified a number of possible solutions. Again, there are fewer possible solutions here than there were problems. And they don't necessarily address all the problems.

Possible Administrative Issue Solutions

- Encourage organizations to create policies for handling role changes (RFC 1244)
- Identify needed tools for implementing role handling policies
- Develop tools to configure and maintain system security

System Integration Subgroup November 5-6, 1992

**Slide 37**

**Audit Problems**

One main area that we talked about, apparently like everyone else, was audit. And interestingly enough, we identified some of the same needs; tamper-proof auditing; secure audit trail transport. There are some techniques that are currently available. And so education about those techniques is one of the things that we listed on our set of solutions. There's some need for standardization of those things as well, so you can do auditing across heterogeneous systems more easily.

There are some issues about content of audit trails. For some uses of audit trails, you want lots of stuff; for some uses of audit trails, you want very small, concise alert-type messages. Whether those should be mixed or separated into different kinds of audit streams is an issue that needs some attention. It's partly a technological issue; it's partly a social and political issue. There is a need for being able to integrate audit trails from application programs, not just systems. It's a need for a system-level mechanism that will support application level auditing.

And then there are other issues, like the availability of intrusion detection software. There should be more of it, in other words. And more technological issues like global synchronization of time. Again, things for which there may be solutions, or partial solutions, available, but not in a widespread use.

Audit Problems

- Tamper-proof auditing
- Secure audit trail transport
- Canonical audit trail format
- Canonical audit trail content
- Balancing the needs of privacy and auditing
- Application-level auditing
- Intrusion detection package should be available
- Time synchronization-global

System Integration Subgroup November 5-6, 1992

**Slide 38**

**Possible Audit Solutions**

Here are things that can be done. The first three bullets on this list could be thought of as topics for possible future workshops. Audit is a big thing. I think it is probably too big an issue to sponsor a single workshop on and cover very much ground. It needs to be broken up in to smaller pieces.

Possible Audit Solutions

- CERT sponsor creation of list of needed audit data for various sources (e.g., OS, applications, sub-systems)
- Development of canonical audit trail format (TSIG efforts?)
- Development of a standardized real-time audit trail transport mechanism
- Continued development and deployment of intrusion detection software
- encourage use of existing technology (e.g. time synchronization, WORM drives for audit data)

System Integration Subgroup November 5-6, 1992

**Slide 39**

**Cost of Security Problems**

Problems in . . . . This headline doesn't read quite right. It's not the cost of security problems, but it's problems related to the cost of security. These are things like old systems around, still around, that don't do things the way we would like to do them in our current networks, but we have to have them for various reasons: people won't change; or have old applications that still have to be run and only run there; things of that sort.

And the problem [that] people simply don't want intrusive security measures and won't accept them in many cases.

Related to the Cost of Security Problems

- Legacy systems
- Users don't want intrusive security: e.g. tokens, frequent password changes, machine generated passwords

System Integration Subgroup November 5-6, 1992

**Slide 40**

**Possible Cost of Security Solutions**

Potential Solutions. This one we had a much longer list. [quiet for reading]

Possible Cost of Security Solutions (?)

- Education
- Policy
- Management Mandate
- Perceived status on benefit (e.g. free calculator in token cards)

System Integration Subgroup November 5-6, 1992

**Slide 41**

**Possible Cost of Security Solutions (2)**

[more quiet for reading] There are some very long-term items here too. Raising the standards of due care, for example, is a legal and social issue, which is not going to be short- or medium-term.

Possible Cost of Security Solutions (2)

- Legal requirements
- Raising the standards of “due care” (commonly accepted industry practice)
- Transparency of controls
- Simply upgrade
- Ostracize (for sites / nodes that refuse to upgrade)
- Replace old application
- Planned obsolescence (phase out support)

System Integration Subgroup November 5-6, 1992

**Slide 42**

**Policy Problems**

Problems in dealing with policy. There are several kinds of issues here. The problem of comparing policies on different machines, with different systems. The problem of determining what your overall policy is when you’ve integrated systems. The difficulty of stating the policy, the policies, that cover all the issues. Those are important, which is at least confidentiality, integrity, denial of service, and privacy. Getting all that folded into a policy statement is difficult.

An issue we talked about quite a bit this morning was the issue of how much privacy a user can expect on a system. That's currently a hot topic at the Justice Department, but there are several unknowns there that need further attention.

Policy Problems

- Hard to compare security policies implemented on different OSs
- Hard to state security policies
- Privacy issues are not considered
- What privacy a user may expect is unknown

System Integration Subgroup November 5-6, 1992

**Slide 43**

**Possible Policy Solutions**

Possible Solutions. Pay for more research. That's always a solution. One we'd all vote for.

The second one here refers to the idea of encouraging management to at least address how much privacy users can expect in the systems for which management is responsible by including such a statement in their corporate policy. This will not solve the problem of the expectation of privacy, but it will at least protect management a little bit more than the current policy.

Possible Policy Solutions

- Fund research (fully!)
- Address Privacy issues (including disclosure to users)

System Integration Subgroup November 5-6, 1992

**Slide 44**

**Vendor Issue Problems**

We had a number of integration issues related to vendors. Many of them centered around what vendors are and do. They have proprietary rights—they want to keep them that way. They advertise to the edge of the envelope, and beyond sometimes. They deliver things frequently with all of the security mechanism turned off, because many of their customers don't care yet.

Vendor Issue Problems

- Proprietary algorithms (e.g. RSA encryption)
- Vendors won't play
- False advertising by vendors
- Secure installation defaults

System Integration Subgroup November 5-6, 1992

**Slide 45**

**Possible Vendor Solutions**

We're getting down to the bottom of the categories. And we had fewer potential solutions that might be useful. These in particular all address the false advertising issue which is the least technological of all the issues, I think.

Possible Vendor Solutions

Note: These solutions primarily address the problem of false advertising

- Education
- Consumer Reports by an independent organization (to include ease of use of security features). Perhaps use an existing publication.
- Food and Drug Administration model of strict enforcement
- Legal approaches. Truth in Advertising. Litigation.
- Glossary of common definitions
- CERT influence on vendors

System Integration Subgroup November 5-6, 1992

**Slide 46**

**Standardization Problems**

We identified a number of things where standardization might help. That is, the use of encryption. The lack of standardization in assigning TCP/IP ports. This makes firewalls difficult to implement, I am told by my friends.

The last bullet [really] says the DoD is still behind on what networks do and should do.

Standardization Problems

- Encryption standards
- Lack of standard tcp/ip ports (e.g. for firewalls)
- DoD guidance doesn't work for client-server networks

System Integration Subgroup November 5-6, 1992

**Slide 47**

**Possible Standardization Solutions**

We didn't have any standardization solutions, other than work on standardization. The obvious one.

Possible Standardization Solutions

- (none)

System Integration Subgroup November 5-6, 1992

**Slide 48**

**Other Problems**

Then we had a catchall for "other" problems.

We decided secure unitary login (that is, one login for the whole network) might or might not be a good idea. There might or might not be good ways to do it, but it definitely is an infinite sink for discussion. *So* we put it in the "Other" list.

And then this last bullet: "Mutual Authentication of Distributed Hosts." This is the same issue that other groups talked about in somewhat different terms. Mutual authentication of distributed hosts may or may not be hard. It is certainly not done. It would probably be helpful.

Other Problems

- We need secure unitary login
- Mutual authentication of distributed hosts is hard
- Hiding assessment details while publishing assessment results

System Integration Subgroup November 5-6, 1992

**Slide 49**

**Possible Other Problem Solutions**

And again . . . . This was at the end of the set of categories—the hard problems.

Questions?

[There were none.]

Possible Other Problem Solutions

- (none.)

System Integration Subgroup November 5-6, 1992

**Slide 50**

# REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS
None

2a. SECURITY CLASSIFICATION AUTHORITY
N/A

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for Public Release
Distribution Unlimited

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE
N/A

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
CMU/SEI-93-SR-20

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Software Engineering Institute

6b. OFFICE SYMBOL (if applicable)
SEI

7a. NAME OF MONITORING ORGANIZATION
SEI Joint Program Office

6c. ADDRESS (city, state, and zip code)
Carnegie Mellon University
Pittsburgh PA 15213

7b. ADDRESS (city, state, and zip code)
HQ ESC/ENS
5 Eglin Street
Hanscom AFB, MA 01731-2116

8a. NAME OFFUNDING/SPONSORING ORGANIZATION
SEI Joint Program Office

8b. OFFICE SYMBOL (if applicable)
ESC/ENS

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
F1962890C0003

8c. ADDRESS (city, state, and zip code))
Carnegie Mellon University
Pittsburgh PA 15213

10. SOURCE OF FUNDING NOS.

| PROGRAM ELEMENT NO | PROJECT NO. | TASK NO | WORK UNIT NO. |
|---|---|---|---|
| 63756E | N/A | N/A | N/A |

11. TITLE (Include Security Classification)
Results of a Workshop on Research in Incident Handling

12. PERSONAL AUTHOR(S)
Thomas A. Longstaff

| 13a. TYPE OF REPORT | 13b. TIME COVERED | 14. DATE OF REPORT (year, month, day) | 15. PAGE COUNT |
|---|---|---|---|
| Final | FROM TO | September 1993 | 60 |

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

| FIELD | GROUP | SUB. GR. |
|---|---|---|
| | | |
| | | |
| | | |

18. SUBJECT TERMS (continue on reverse of necessary and identify by block number)
computer emergency response, incident handling, Computer Emergency Response Team, CERT, SEI, security engineering, computer security, computer security research

19. ABSTRACT (continue on reverse if necessary and identify by block number)

This document contains the results of the first CERT[sm] Invitational Workshop on Research in Incident Handling, held at the Software Engineering Institute in November 1992. The workshop was convened to address a wide spectrum of computer, network, and information security topics from the perspective of incident handling, both in the present and in the future. The intent was to bring together researchers, incident handling specialists, users, system administrators, and managers to encourage an exchange of information and experience. Specifically, it was intended to identify lucrative areas for research and development in improving the practice of incident handling and in applying the experience- and information-base that the CERT Coordination Center has amassed during its existence.

(please turn over)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ■ SAME AS RPT ☐ DTIC USERS ■

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified, Unlimited Distribution

22a. NAME OF RESPONSIBLE INDIVIDUAL
Thomas R. Miller, Lt Col, USAF

22b. TELEPHONE NUMBER (include area code)
(412) 268-7631

22c. OFFICE SYMBOL
ESC/ENS (SEI)

ABSTRACT — continued from page one, block 19