DARPA Research Panel 2:

Secure Networking and Assurance Technologies

Panel Chair: Teresa F. Lunt, DARPA

Panelists:
Karl Levitt, UC Davis
John McHugh, Portland State University
Steve Kent, BBN
Gary McGraw, Reliable Software Technologies
Doug Weber, Key Software
Lee Badger, TIS

Today's security solutions are being built for aging computing and communications technologies. Many of these solutions will not scale to the technologies of the future, and the future is just around the corner. For example, mechanisms that depend on cryptographic authentication of every packet in a data stream, that require frequent reference to distant directory servers to ascertain certificate validities, and that require lengthy appendages of signed certificates, may not be able to keep up with the speeds of new high-speed networking technologies or be at all appropriate for mutually authenticating software agents. Access control policies that were designed for a closed environment may not scale well to world-wide-web-style environments in which there are frequent interactions between unacquainted entities, or to a highly networked environment in which new alliances are quickly forged and terminated. The new phenomenon of cyberspace opens up privacy concerns that were not present in small, closed communities where one's every computing activity was not on display to the entire world.

In many cases, the focus on security must change from the individual end system to the network. For example, in intrusion detection, we must find analysis techniques that scale to very large systems (i.e., that do not require massive amounts of data to be collected) and that can produce reasonable results with partial data (since not all portions of a network are always visible). These systems should also be refocused to monitor network activity rather than exclusively end-system activity, and they should be made to work in a variety of networking technologies. We must better understand how to instrument our systems and networks so as to give us the requisite visibility. We also need to develop both intrusion detection and system management tools that can operate across administrative domains, or that may work with a network of other autonomous detection or management systems in a cooperative or hierarchical manner. These systems should be capable of dealing with extensive heterogeneity both with respect to the systems monitored and the detection and management systems themselves.

Most of the information infrastructure is going to be with us for a very long time. Telecommunications systems, electric power generation and distribution systems, financial systems, and transportation control systems will slowly evolve but will retain their legacy character through generations of technology improvements. In addition, many new critical systems, such as medical devices, defense command and control systems, and nuclear power plant control systems, are being constructed using commercial software products. We must begin work now to understand and deal with the risks of using commercial and legacy components in systems we depend on for our national well-being and personal safety.

We need strategies for working around the problems that are inevitably to be found in legacy and consumer-quality products. We need architectural "workarounds" to augment the strengths or compensate for the weaknesses of these components. DARPA is investigating whether security can be introduced into a system by developing security "wrappers" for certain system components. With this approach, wrappers would be used to introduce certain security functionality without altering the legacy code or the other system components that use it. The idea is to gain control over specific interfaces where a security function can be inserted. Such interfaces could be library calls, system calls, or other interfaces internal to a subsystem. For the approach to have any validity, it must be possible to ensure that all input to and output from the wrapped component can be intercepted by the wrapper; in effect, the wrapper becomes a reference monitor for the policy it enforces. This is the fundamental new assurance question for the approach.

This new approach requires new theories of secure composition of a system from components (including wrappers) and technologies for security integration. We must broaden the types of analysis that can be performed far beyond such narrow considerations as secure information flow for multilevel security. We must reason, for example, about how such diverse aspects of security as authentication, access control, and encryption contribute to overall system security when inserted into a system in various ways. In addition, our reasoning must allow for ignorance, empirical properties, or worst-case assumptions about legacy components. To support such reasoning, we must adequately specify the components; research is needed in order to understand what must be specified.

Security can be inserted in this manner to meet a variety of objectives. For example, it is easy to imagine how a wrapper could impose an access control policy on the wrapped component, or encrypt the outputs and decrypt the inputs of a components, or perform inter-component authentication, or perform message filtering. One could also design these wrappers to add security monitoring and intrusion detection capability. Ideally these wrappers should be designed so that the specific security solution is a modular part of the wrapper. This would allow the module to be replaced, for example, when it is desirable to use a stronger security solution. This should also allow multiple security modules, enforcing orthogonal policies, to be inserted in the same wrapper.

It has long been held by the security community that security must be designed into a system from its inception and cannot be added on later; we must investigate the feasibility of this new approach and discover how far and for what aspects of security it can be made practical.

The panelists explore these and other issues being investigated in the DARPA research program.

Secure Mobile Networks
John McHugh, Portland State University

Very little work has been done to integrate security and network-layer mobility into real systems that tackle the issues of secure enclaves. The work that we are undertaking will result in the development of a high performance Secure Mobile Network and insights into its use as part of the National Information Infrastructure.

Our goal is to produce a system that supports the establishment of secure enclaves or secure virtual networks among mobile workstations. We intend to combine a secure metwork layer including network layer authentication and encryption with robust Mobile-IP networking allowing secure mobility. Two-way tunnels will be used to allow remote networks or hosts to join a secure network across insecure topologies. We will investigate and design solutions for distributed access control protocols, and redundant systems needed for overcoming the single point of failure problems in the current Mobile-IP architecture.

In general, the IP community has limited experience with network layer security. Network layer security must be integrated with wireless Mobile-IP, another area in which the community has limited experience, and with other mechanisms needed to provide a suitably rich architectural environment that will deal with access control and other security issues as well as redundancy and other reliability issues. In attacking these problems, we will follow a rigorous engineering approach, guided by appropriate formal methods. We believe that protocols used in this sphere should be formally analyzed and their implementations subjected to rigorous software engineering techniques. Many network security problems are due either to faulty protocols or to flawed implementations or both and we hope to avoid these problems in our work.

Our initial system will combine a secure network layer, with Mobile-IP and two-way tunnels. A secure network layer has an operating system architecture component and a protocol component. For protocol components, we are following the IETF IPSEC working group recommendations as closely as possible in order to maximize the potential for technology transfer. Our protocol will provide authentication and encryption at the network layer.

The network architectural component includes access control and key management subsystems at the network layer. Outward and inward bound packet addresses will be looked up in the access and key management tables and appropriate actions, encryption, etc., will be taken. Access and key management daemons (application-layer processes) will allow for higher-level protocols and information exchange. We will design and implement a distributed access-control protocol. Such a protocol is analogous to current intra-domain routing protocols such as OSPF or RIP where clean separation of policy and mechanism exists between daemons and IP-level lookup tables.

Network layer security will be integrated with a Mobile-IP network architecture. The Mobile-IP architecture consists of a routing infrastructure containing three kinds of entities: Home Agents (HA), Foreign Agents (FA), and Mobile Nodes (MN). A single organization's MNs will typically belong to one or more IP subnets where the subnet address is topologically local to the organization. The HA is in charge of routng packets from the rest of the network to the MNs and tracks each MN via a registration protocol. When an MN moves from its home to a foreign subnet (or from one foreign subnet to another to another), it will send a registration packet to the HA via the current FA, which acts as a cell manager. After registration, the HA can forward incoming packets to the MN by encapsulating them in an outer IP wrapper with the FA as the destination. This is referred to as a "tunnel".

Currently, Mobile-IP assumes tunnels go one-way only from the HA to the FA. A recent CERT advisory has pointed out the dangers of local network addresses crossing from the outside to an inside network via a firewall. This appears to be a generic flaw in Mobile-IP and would prevent mobile systems from talking to local systems across current firewalls. We suggest that tunnels may be used as network bridges to allow remote mobile routers or hosts to convey their packets back across an insecure network to a secure router, thus forming a secure virtual network.

In addition to building an integrated secure mobile network that allows secure enclaves, we propose to investigate protocols that allow redundant Home Agents and Foreign Agents. Protocols that allow registration, handoff, and exchange of information between Home Agents are needed. A successful attack on a Home Agent or its failure for any reason could mean the catastrophic loss of a mobile network. A protocol for server redundancy should allow the mobile system to support more than one Home Agent.

Redundancy of FAs is also an important topic, since loss of a local FA might mean loss of communication with home or worse, complete loss of communication within a local cell. IP as currently construed assumes that the Address Resolution Protocol (ARP) cannot be used to establish communication between two hosts that are on the same link but are on different IP subnets (RFC 1122). Communication must be done through a router on the link (in Mobile-IP terms, the router would be the FA). We propose to develop an ad hoc protocol that would allow hosts within the same link to communicate directly where possible. Topologically, example systems could comprise a small mesh in which any system can address all other systems or a daisy chain in which each system can only address one or two other systems. It is always possible that systems might be able to talk to one system and not reach another; "can communicate with" is not transitive for radio.

Resolution of the routerless routing problem is a key factor in facilitating ad hoc networks. We want to be able to create these anywhere two or more MNs can communicate, whether or not a HA or FA is reachable.

We have established a Mobile-IP infrastructure in two buildings on the PSU campus. There are three agents (1 Home Agent (HA), 2 Foreign Agents (FA)) in our PCAT engineering building and one Foreign Agent in the Mill Street CS Lab building. Three graduate students, 4 professors (3 CS, 1 EE) and two staff members have mobile laptops. These run on a slightly modified version of the Free-BSD operating system.

In addition, we have established FAs at two off campus sites using modem connections via SLIP or PPP to connect to the campus network. In doing this, we have essentially managed to take PSU IP addreses to remote, disjoint locations. This allows Mobile-IP to be used to implement disjoint networks without requiring that internal routers actually know or support additional routes. It appears that this may permit a more efficient implementation of IP address space.

We have implemented a simple, but effective timestamp mechanism that counters most replay attacks while preventing replays from being used as a denial of service attack.

By the time of the conference, we hope to have made additional progress on several fronts. Our Mobile-IP implementation (MN, HA, and FA) will be available to interested parties by the first quarter of FY97. Check our web site for details (http://www.cs.pdx.edu/research/SMN/).

We are starting to integrate IPSEC with our Mobile-IP implementation, using Fortezza cards being supplied as GFE to rpovide encryption support. We will complement these with software encryption and possibly DES hardware encryption for the nodes for which we do not have Fortezza cards.

We will expend significant efforts toward making Mobile-IP more robust and secure through the provision of redundancy. There are three areas of work: 1. ad hoc routing, i.e., how MNs can route amongst themselves and also find paths to agents through other MNs; 2. redundant FAs; and 3. redundant HAs.

Adaptable Dependable Wrappers
Doug Weber, Key Software

The Adaptable Dependable Wrappers project is exploring a flexible way to build dependable distributed systems from software components. We are designing a prototype toolkit for generating adaptable dependable wrappers for the components of a system. We intend to test the flexibility of our approach by implementing the toolkit and using it to generate some sample distributed applications.

A *wrapper* for a software component forms a boundary layer between the component and all other components that interact with it. The purpose of the wrapper is to translate and filter the view these components have of each other's behavior.

A *dependable wrapper* imparts critical properties to each component that it wraps. For our purposes, "dependability" includes both survivability and security. Some dependable wrappers have been built before, but without the flexibility of our approach. A survivable wrapper typically wraps a group of replicas of the component, coordinating the replicas for fault tolerance. Security wrappers have been used for many purposes, including authentication and access control.

We are generalizing this previous work by creating dependable wrappers that are also *adaptable*. We mean "adaptable" in a general sense, including both *configuration* at compile time and *reconfiguration* at runtime. An engineer will configure a dependable component wrapper framework at compile time by choosing from a library:

- algorithms and protocols that support critical properties he specifies;
- a design that will work efficiently in the component's environment.

At runtime the wrapper will reconfigure itself automatically when it interacts with other components. An adaptable dependable wrapper:

- can learn the specification of another component;
- can decide whether the other component's specified critical properties are sufficient to support its own;
- can decide whether to trust that the other component actually implements its specification;

- can learn from the other component new protocols that must be used to guarantee critical properties;

- offers information about its own properties to other components.

Adaptable dependable wrappers offer the following advantages over existing technology:

- The wrappers can be used to gain security and survivability in a wide variety of distributed systems. Components can be wrapped specifically to support each system's requirements.

- A component of a long=running system can be replaced (for modification, upgrade, or with a new application) without restarting the system. Replacement is easier and arguably safer than in current distributed systems because a new component teaches others about itself.

- A survivable system can degrade gracefully after massive failures by weakening its dependability specifications. The surviving components may be able to continue functioning by learning to interact with new, less dependable, components chosen from a larger pool.

The Adaptable Dependable Wrappers project is part of DARPA's Information Survivability program.

Generic Software Wrappers for Security and Reliability Lee Badger, TIS

Very large-scale information systems are increasingly built by combining numerous independently developed software components. Components may be programs, linkable code libraries, and, increasingly, network applets based on emerging software frameworks (e.g., CORBA, OLE, CGI, Tcl, Java). While use of independent, and standardized, components reduces cost, component failures and unintended interactions among components seriously threaten the reliability and security of information systems that use them. Components are often engineered for "commercial" assurance but then are deployed within critical systems requiring high assurance. Of particular concern are network applets that bring new power to rapidly deploy information systems but also add risk: applets often exchange interpreted data, which makes them highly vulnerable to corrupted data. Applets may also be dynamically reinstalled: this potentially exposes information systems to flaws in future as well as current software components.

Dramatic advances in information system security and reliability will require techniques both for limiting the damage that can be caused by individual components and also for adding reliability features tailored to system mission requirements. A variety of techniques (e.g., Internet firewalls, extensible operating systems, fault isolation) control or enhance component interactions, but these techniques are too costly, not generic, or provide inadequate support for coordinating security and reliability policy data.

This project will develop techniques and tools for specifying and implementing generic software component wrappers. Generic software wrappers will intercept component interactions and bind them with additional functions that implement practical security (e.g., restricting, filtering) and reliability (e.g., redundancy, crash data recovery) policies. We believe that a successful wrapping technology must: 1) wrap existing components, 2) accommodate a large number of software interfaces and policies, 3) work in numerous execution environments, 4) be optional and consistent with high performance, and 5) be capable of high assurance.

This project will develop a prototype Wrapper Development Framework to demonstrate practical software-wrapping technology that meets these criteria. The wrapper development framework will include a Wrapper Definition Language (WDL), a Generic Wrapper ToolKit, a Wrapper Support Interface, and two systems that implement it: a wrapper-supporting UNIX prototype and a wrapper-supporting Java prototype. The Generic Wrapper ToolKit will implement wrappers expressed in WDL and will provide tools to wrap and unwrap selected components at runtime. The Wrapper Support Interface will define a modest level of generic wrapper support (necessary for high assurance) suitable for standardization and inclusion in mainstream execution environments.

This project will implement wrapper support in both a kernelized UNIX and an interpreted Java environment to build confidence that the approach is general and that WDL wrappers are portable. By demonstrating practical, generic software-wrapping technology, this project seeks to provide a basis for significant security and reliability increases in large-scale information systems based on reusable software components.