DARPATech 2004
Security-Aware Critical Systems

Mr. Lee Badger
Information Processing Technology Office
Defense Advanced Research Projects Agency

We know that the United States depends critically on reliable and secure information systems. But our systems are far from secure. They are vulnerable and, from a security perspective, brittle.

Perhaps even more important, military software consumers have no realistic way to view and understand the security of the systems they use, or to measure it quantitatively. DARPA is committed to making software and operating systems of the future far less vulnerable to attack, and far more capable of explaining their security capabilities and limitations to their users.

We're going to make critical software systems much smarter about their security properties. We intend to shift the burden from people to systems and to a new generation of intelligent software development and analysis technology for assured operation in hostile environments.

For a moment, imagine a system that dynamically learns about the threats in its environment, that forms hypotheses about which threats might exercise vulnerabilities, and that automatically adjusts to immunize against those threats. Imagine a system that presents a moving target to adversaries, so adversaries can't carefully refine their attacks in advance. Imagine a system that is flexible enough to adapt to new threats and military needs without the help of an army of programmers and system administrators. Imagine a system that is aware of the risks that go along with advanced system functions, and can provide a coherent summary of risks, benefits, and status in a form that is most useful to the consumer, human or machine.

We have no choice. Ever greater complexity is driving ever greater functionality and ever greater vulnerability. DARPA is committed to addressing emerging vulnerabilities by building systems that naturally extend to augment security, that are aware of their vulnerabilities, and that have quantitatively measurable security properties.

The DARPA Dem/Val program is building an intrusion tolerant system that can stand up to sustained expert attack. It uses coarse-grained redundancy and diversity to protect itself while providing a critical military service.

Now, folklore has it that when we stand up a complex system against a skilled red team attacker, the system loses. In early 2005, at Rome Labs, we're testing the folklore. We are pitting this system, developed by BBN, against two expert red teams. We're giving these red teams complete knowledge of the system's design and implementation, and 12 hours to subvert it. If the system survives, that will be news. But either way, we will learn much about how to keep a system's critical functions operating for a limited period in a very hostile setting.

For many military applications, though, critical functionality must be provided, even in hostile environments, for much greater periods of time. The critical functions may be fixed in advance, but the systems themselves must dynamically learn about their environment, and must self repair in order to assure that their most important functions remain available.

The Self-Regenerative Systems program is developing new techniques for building systems that, over time, improve their ability to deliver service through a process of learning and immunity building. This research, just starting, will develop new techniques for generating diversity in computing systems, for building immunity against cyber attack, for managing substantial reserve resources, and for analyzing the actions of possibly-malicious insiders.

Our current research initiatives begin with the realization that perfect software is not a realistic ambition—there will always be flaws so we must operate securely in spite of them; we must compensate for them. But compensation is expensive and does not apply in all the settings where we need dramatic security improvements. For instance, we need high security in emerging resource-constrained environments like sensor nets and unmanned vehicles.

We need security in high-performance servers. We need security in systems that are being updated frequently and where the security requirements are themselves evolving rapidly and there is little time to puzzle over subtle interactions.

While security for systems is critical, we also need security in the frequently-updated applications that run on our systems and protection of application-layer entities such as Web pages, radar tracks, maps, and instant messages.

As Ron said, we need security for emerging cognitive systems. And we need practical and timely ways to understand and measure the security that our systems do in fact provide.

Many security problems can be traced to a failure to properly set the access rights granted to application programs. For example, a worm can't spread if it can't access another system that has the flaw it exploits. A virus cannot spread if it can't modify a boot sector or write to vulnerable files. A compromised server cannot launch a denial of service attack if its access to networking is properly mediated. Controlling such accesses does not prevent program failure, but it limits the impact and is a necessary building block for trustworthy military systems.

Theories of access control exist, but bridging the gap between theory and practice has been extremely hard. Many access control mechanisms have been invented, for example, access control lists, multi-level security, and type enforcement, but they go mostly unused.

At its heart, the problem is complexity: complex applications that we can't understand well enough to regulate, and complex systems that become incredibly brittle as soon as we try to extend them in any way not explicitly anticipated by their designers. This problem is exacerbated by the fact that our systems are not static: they continue to evolve and our solutions must accommodate that evolution.

We now seek practical technology to move from non-extensible systems to extensible systems that bridge the complexity gap between theory and practice. We seek technology to build Security-Extensible Systems that are optimized for minimizing the level of necessary privileges

granted to application software, and for making the management of security policies a structured process amenable to reasoning and assurance-building. To be practical, this technology must be lightweight with a small footprint, and work in resource-constrained settings.

We seek an operating system base that realistically supports fine-grained security policies and that allows us to adapt them, with high confidence, to changing needs. We seek architectures that allow security extensions, or interpositions, to be dynamically added into systems and to regulate their behaviors with high confidence. To do this, we need systems that make it possible for security observer components to understand the interactions they must control, an impossibility with current systems.

We seek a system that fundamentally provides visibility into resource accesses and application interactions so that the inevitable future additions can be made without guesswork. If achieved, a Security-Extensible system would reduce damage from failing software components, but it would not prevent their failures.

Because we are interested in applications in which compensation mechanisms are not feasible, we need other ways to mitigate the effects of failure that do not involve the illusion of perfect software. We have no alternative but to manage the residual risks of our flawed systems, but our systems have no concept of risk.

If you look through the commands available on your computer, you are unlikely to find one that tells you how likely the others are to work. You are even more unlikely to find one that tells you the environmental factors under which different commands are not trustworthy, and the probability that those factors are present.

We seek technology for building software components that can introspect about their security properties, can dynamically adjust to new requirements, and can explain their capabilities and anticipated limitations in a usefully-broad variety of forms, including programmatic interfaces and structured natural language.

Instead of recovering from certain damage, or perfectly avoiding damage, we seek to build software that uses self knowledge, descriptions of the threat environment, and reasoning to minimize the likelihood that potential weaknesses will be exploited during the duration of a mission.

Our goal is to make security self-monitoring, security flexibility, reporting, and reasoning organic features of software components, and to build systems that reason about risk. For example, security-aware components will know, among other things, which of their interfaces are trusting and possibly vulnerable, what code paths have been subjected to what kinds of analysis, which parts of their design and implementation are relatively mature or immature, the history of past exploits, a description of critical mission requirements, and a threat model.

Security-aware components will be able to dialog with managing entities, whether human or machine. Building security-aware software raises a number of challenges. For example, how can security policy and status be summarized so that the descriptions produced can be understood while at the same time not omitting critical details.

The technology may substantially increase software complexity; how can this be done without introducing new vulnerabilities? Addressing these challenges and others won't be easy, but it will almost certainly be easier than building perfect systems.

We can't achieve perfection, but we may be able to build agile and reflective systems that can reason about their exposure in hostile environments, and that can present a useful summary of security choices without drowning managing entities in an ocean of extraneous details.

By building security awareness into software components, this initiative seeks to make the most useful security information available to decision makers, and to give decision makers the greatest opportunity to avoid unnecessary risk and to understand necessary risk.

Any success in security-extensibility and security-awareness will make military systems more secure, but without further research we will still not know how much more. At present there is no way to quantify information assurance measures for critical properties such as fault tolerance, intrusion tolerance, confidentiality, integrity, or robustness. As a consequence, we have difficulty knowing with any level of certainty whether our military information systems will operate during a conflict.

We seek to develop practical, quantitative measures of information assurance in military systems; such measures could enable construction of military systems with guaranteed levels of protection against cyber attacks. Assurance is a widely-used but misleading term. In reality it refers to a person's belief about the future behavior of a mechanism, based on evidence.

We seek to develop technologies to understand and analyze the critical behaviors of our systems and the kinds of evidence that can be generated to build confidence that our systems will behave as desired during a conflict.

We already know a lot about the kinds of behaviors that are desirable, such as security and survivability, but we are hard pressed to quantitatively measure them. Recent developments, though, suggest that progress is now possible. First, the steady accumulation of techniques such as model-based checking, attack trees, and attacker intent recognition, have not been tried, combined, and measured in a comprehensive way. Second, multiple publicly-available historical databases of real-world exploits are rapidly growing and can be mined for insight on real-world assurance failures, and successes.

We seek to combine these developments, research the resulting theoretic aspects of information assurance; and develop metrics to characterize central security dimensions, and show the relevance of the theory by applying it to a realistic exemplar system. These technology thrusts: security-extensibility, security-awareness, and assurance measurement; will extend our existing work in intrusion tolerance and self-regenerative systems.

Through them, we aim to create new military information systems that will remain useful and trustworthy in chaotic environments and continue to help us cut through the fog of war, even when we are faced with a serious digital adversary.

Thank you.