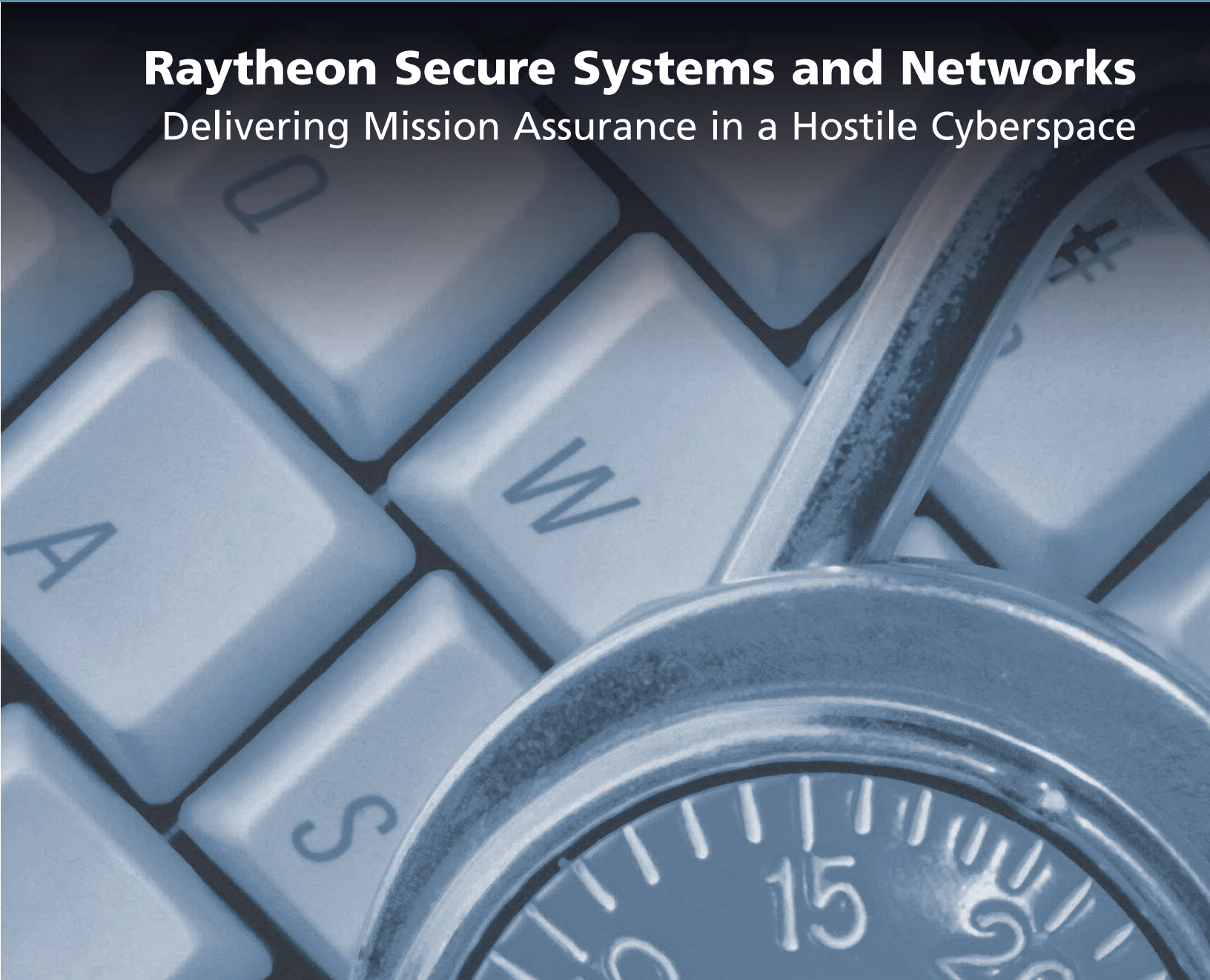


# Technology **Today**

HIGHLIGHTING RAYTHEON'S TECHNOLOGY

2007 Issue 2

## **Raytheon Secure Systems and Networks** Delivering Mission Assurance in a Hostile Cyberspace



**Raytheon**

*Customer Success Is Our Mission*

# Intrusion-Tolerant Systems

**A**s a nation, we need information systems that continue to operate in the presence of a sustained cyber attack. Our systems cannot afford to lose their availability, confidentiality or integrity when an attack becomes an intrusion — that is, when an attack successfully penetrates a system's security mechanisms to form a malicious fault. The need for a system that can tolerate malicious faults, deemed "intrusion tolerant," is based on the reality that some attacks will inevitably succeed, and therefore must be tolerated without compromising system integrity.

Today's systems are not intrusion tolerant, as security mechanisms can only prevent or detect some intrusions. Because of this limitation, a system may fail to perform its mission when an attack is successful, and it may be unable to recover quickly, if at all. What's more, it may fail to detect an intrusion that compromises its confidentiality or integrity. Clearly, if today's systems are to deliver Mission Assurance in the face of information warfare, they need to be more secure than they are now.

As a result, Raytheon is currently working to develop an architecture for intrusion-tolerant systems. This work leverages the results of recent DARPA programs that have developed and demonstrated intrusion-tolerant technologies and architectures. Raytheon has participated in one of these programs (Self-Regenerative Systems) and is now working with the research community to apply technologies and concepts from these programs.

## Current and Future Systems

Intrusion tolerance takes survivability to a new level. While today's systems prevent most intrusions by blocking *known* attacks, intrusion-tolerant systems must handle

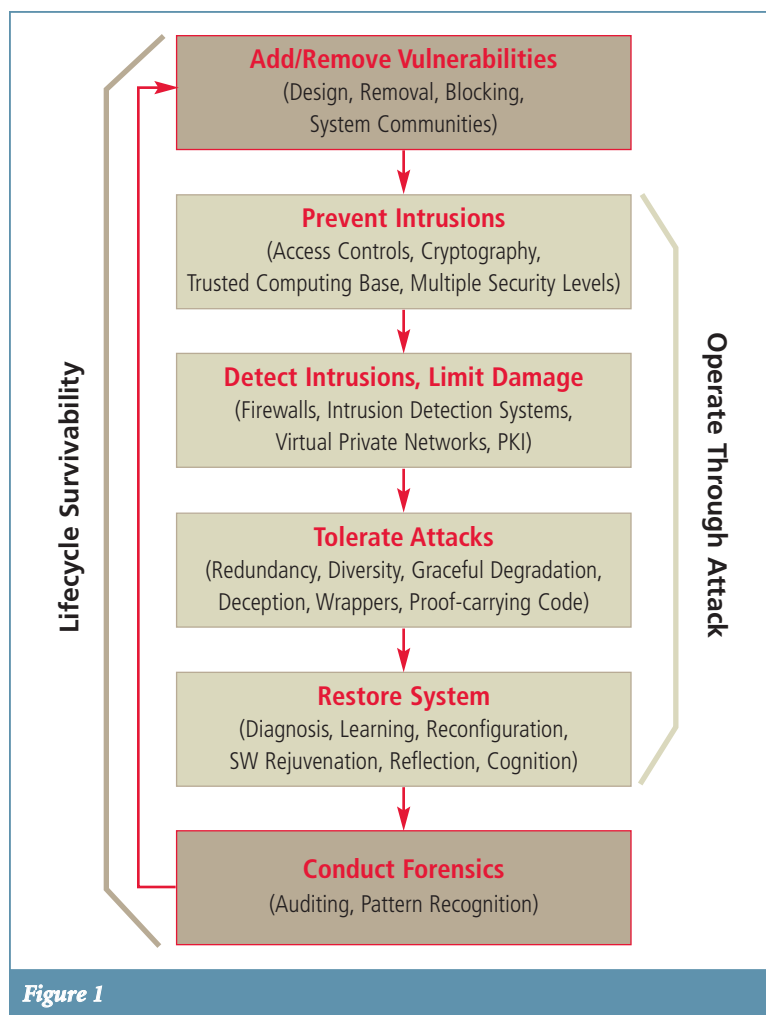
*unknown* attacks. It is not enough just to detect intrusions; a system needs to decide on a course of action that will effectively respond to the attack. Data from multiple sensors must be correlated in order to better diagnose, isolate and respond to attacks. Today's responses usually involve human diagnosis and interaction, which is slow and often inaccurate. To handle varying attacks, operating scenarios and prevent damage, diagnosis and response need to be automatic, adaptive and at machine speed.

an attack by gracefully degrading its level of service and its non-critical functions as needed. It will recover its full functionality and level of service automatically after the attack.

Looking farther into the future, we can expect systems to reason about attacks, develop more effective responses to new attacks, and improve their survivability over time by identifying and removing vulnerabilities. This idea is illustrated in the "Lifecycle Survivability" flow in Figure 1. In addition,

networked systems will share their insights with one another, so that whole families of similar systems can rapidly gain immunity from new attacks and remove their common vulnerabilities.

Automating vulnerability diagnosis and removal will make lifecycle survivability improvement practical. A system's survivability naturally tends to degrade during deployment, as attackers discover its vulnerabilities and new attacks emerge. Today, vulnerability diagnosis and removal are complex, manual time-consuming activities, creating lengthy vulnerability windows during which vulnerabilities can be continually exploited. This is a common problem today among systems connected to the Internet.



In addition to blocking and detecting most intrusions with mature security technologies, an intrusion-tolerant system will use new generations of security technologies to tolerate the intrusions that penetrate these defenses. This idea is illustrated in the "Operate Through Attack" flow in Figure 1. An intrusion-tolerant system will respond to

Ongoing DARPA research seeks to automate vulnerability diagnosis and removal at the application level. Its goal is to develop a software execution infrastructure that monitors and augments application behavior so that multiple copies of an application behave as a self-aware community. In turn, this community would collaboratively

diagnose attacks/bugs/errors; generate appropriate configuration changes, patches, filters, etc.; and generate a community-specific situation awareness gauge that predicts the likelihood and timing of imminent problems. Eventually, this will lead to automation at the system level.

## Architecture Principles

Intrusion tolerance cannot be achieved by simply adorning a system with security technologies after it has been designed. A system's architecture must support intrusion tolerance as well. A number of architecture principles apply:

- The architecture should first maximize its intrusion prevention and detection capabilities using mature security technologies and techniques.
- The architecture must tolerate Byzantine failures. This is because malicious faults can asynchronously occur in any replica and yield Byzantine failures.
- Static diversity, or implementing a function in multiple ways, should be used to avoid common vulnerabilities. For example, research has made it practical to automatically generate diverse executables from the same source code.
- Runtime diversity, which implements a function differently at different times, will make it harder for attacks to succeed. For example, a system could be designed to automatically change its configuration from time to time to confuse the attacker.
- Attack isolation and containment will prevent damage from spreading and bind the set of elements that a system must reconstitute after an attack.
- Correlating alerts from multiple intrusion sensors will allow a system to better diagnose, isolate and adaptively respond to each attack.
- Adaptive response will enable a system to respond appropriately to different types of attack.
- Graceful degradation will prevent an abrupt or catastrophic loss of service during an attack.
- Self-regeneration after an attack will automatically restore full functionality

and level of service. Automation will speed the process and make it reliable.

- Architecture should make weak assumptions about the integrity and availability of its operating environment.

## A Common Architecture

The common architecture for survivable systems applies these principles. It is based on a prototype architecture that was demonstrated on DARPA's OASIS (Organically Assured and Survivable Information Systems) program.

A common architecture offers the advantages of repeatable results and economy. The abstract architecture can be the basis for many system designs. Its reusable software components can be used across many systems.

The architecture is transparent to mission applications, making it easier for the architecture to support legacy applications, as well as new ones. These applications must be model-able as loosely coupled service providers and consumers that use pub-sub-query transactions. While the architecture cannot support hard real-time transactions, real-time systems such as radars can be included as mission applications within a larger system that the architecture supports (such as a C2 system). This protects real-time systems from attack if they are not directly accessible from outside the larger system.

The architecture provides concentric layers of protection to mission applications, system operations and system/security management — placing management functions in the most highly protected zone. These zones are replicated in a Byzantine fault tolerant manner.

A survivable middleware builds security mechanisms on top of a common multicast protocol to enhance integrity, access control, resiliency and graceful degradation. The middleware has redundant protocols and can change its transport protocols dynamically. Session keys and cryptographic credentials are used to manage access con-

trol. Messages, which are checked for valid size, frequency and signature, are briefly held in escrow so that if the publisher appears corrupt, a message is not forwarded. The middleware provides redundant channels that connect each mission application to the core zones of the architecture. If all channels to the core fail, the middleware attempts to attach mission applications directly to one another. Heartbeats are generated by the middleware to indicate that each mission application is alive.

Policy-driven protection "domains" help protect system, process and network components from attack. Domains are used to isolate components, limit their privileges, prevent corrupted processes from accessing critical resources, defend application-specific resources and disallow actions that exceed privileges. Attempts to violate policy generate alerts.

System/security management monitors these heartbeats and alerts. Correlated sensor data helps identify suspicious assets, and contain and isolate attacks. System/security management provides adaptive responses, which are executed by actuators placed throughout the system. Responses can be reactive or proactive. For example, if sensors detect a process's attempt to transition to root, an actuator might kill the offending process (a reactive response). However, if sensors detect file corruption, the system may decide to check and restore files (a more proactive response). If the system determines that a host is compromised it may disconnect the host and reconfigure the system.

## Conclusion

Raytheon is working to take the lead in making intrusion tolerance a reality in defense systems, by engaging the research community and our customers to transition technologies and concepts into working systems. This will make it possible for systems to withstand sustained cyber attacks and achieve Mission Assurance in the face of information warfare. ●

*Tom Bracewell  
bracewell@raytheon.com*

***Do you have a great idea for an article?***

We are always looking for ways to connect with you — our engineering, technology and Mission Assurance professionals. If you have an article or an idea for an article regarding technical achievements, customer solutions, relationships, Mission Assurance, etc., send it along. If your topic aligns with a future issue of *Technology Today* or is appropriate for an online article, we will be happy to consider it and will contact you for more information. Send your article ideas to [techtodayeditor@raytheon.com](mailto:techtodayeditor@raytheon.com). We're waiting to hear from you!

**Raytheon**

*Customer Success Is Our Mission*

Copyright © 2007 Raytheon Company. All rights reserved.  
Approved for public release. Printed in the USA.  
*Customer Success Is Our Mission* is a trademark of Raytheon Company.  
Capability Maturity Model, CMM and CMMI are registered in the U.S.  
Patent and Trademark Office by Carnegie Mellon University.