Defining A Security Architecture For Real-Time Embedded Systems

Tod Reinhart
AFRL-IFTA
2241 Avionics Circle, Suite 32
WPAFB, Ohio 45433-7334

Carolyn Boettcher
Raytheon Space and Airborne Systems
PO Box 902, MS RE/R7/P570
El Segundo, CA 90245

G. Andrew Gandara
Raytheon Space and Airborne Systems
PO Box 902, MS RE/R1/A520
El Segundo, CA 90245

Mark Hama
Raytheon Space and Airborne Systems
PO Box 902, MS RE/R1/A521
El Segundo, CA 90245

ABSTRACT

Providing information assurance (IA) for embedded aerospace platforms in a network-centric battlespace presents new challenges for information-intensive system development and deployment. This paper will discuss ongoing research being conducted by Raytheon under two Air Force programs. As part of this research, Raytheon is assessing the vulnerability of mission-critical platforms to information warfare attacks on the infrastructure required to achieve interoperability and information sharing. This paper discusses Air Force missions, the technologies that are likely to be used to achieve interoperability, ongoing research in IA that can be leveraged, any IA vulnerabilities that are not yet being addressed, and approaches to mitigating those vulnerabilities. Recommendations for promising future research directions are described.

INTRODUCTION

The research described here has been ongoing for four years under the Air Force Research Laboratory's (AFRL) Embedded Information Systems Assurance (EISA) program and is continuing for another four years under the AFRL Secure Interoperability for Real-time Embedded Systems (SIRES) program.

The completed EISA and ongoing SIRES research and technology programs are determining ways to protect information exchange between command and control (C2) and tactical warfighter platforms within a Global Information Grid (GIG). When fully deployed, the DoD-wide GIG will provide a distributed, interoperable infrastructure to enable warfighters to have the right information at the right time.

The GIG was first conceptualized in the DoD Joint Vision 2010, issued by the Chairman of the Joint Chiefs of Staff in July, 1996 [1]. Each of the services subsequently established efforts to develop an implementation of the GIG, i.e.., the Navy's ForceNET, the Army's Objective Force and the AF's Joint Battlespace Infosphere (JBI) [2].

Examples of information that will be available through the GIG includes time-critical targets; intelligence; air, sea, and ground order of battle; and logistics. The foundation of the GIG will be a secure network that enables users immediate access

to data and applications published on the GIG, regardless of their physical location.

Commercial technologies will be used wherever possible in implementing the GIG. As a result, the EISA program concentrated on commercial network-based and middleware technologies that provide secure communication between distributed systems. The SIRES program extends the EISA research to additional middleware and application technologies that are expected to be introduced into tactical and C2 systems in the future to support information exchange.

PROBLEM DEFINITION

During the domain analysis phase of the EISA program, the basic problem definition was established. Under SIRES, the EISA domain analysis is being extended to consider the evolving capabilities of the GIG and the effects of incorporating a tactical aircraft into a GIG warfighting environment. Such an environment is illustrated in Figure 1, where tactical aircraft can access information from or about other aircraft, as well as space, ground, or sea assets.

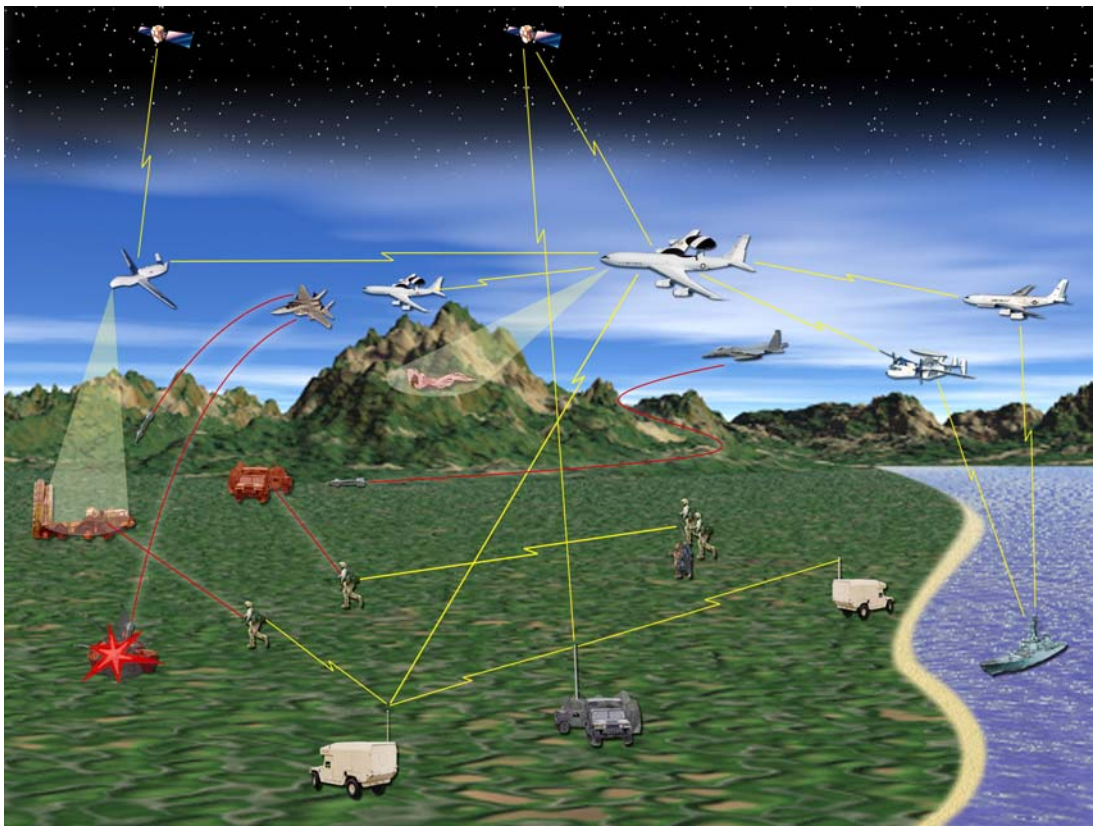

Figure 1 Conceptual Network-centric Battlespace

In this information intensive environment, the tactical aircraft derives several key benefits: increased situational awareness, tight sensor to shooter networks, and dynamic mission planning and redirection. Moreover, the GIG supports the notion of a dynamic environment where tactical platforms can enter and exit the battlespace over the course of a given mission.

However, with this increased information flow among tactical platforms and the dynamic nature of the battlespace, the domain analysis indicated that the tactical aircraft also had an increased vulnerability to passive and active information attacks. To provide information assurance and survivability to the tactical aircraft requires that information integrity must be guaranteed, confidential communications must be protected, and asset availability must be preserved.

As part of the domain analysis, we also looked at trends in military communications. Current tactical datalinks, such as LINK16, have limited bandwidth and are based on specialized protocols and message formats. To meet the communications and interoperability demands of the GIG, it is expected that tactical datalinks will evolve to support higher bandwidths and to make use of higher level, more flexible protocols, such as TCP/IP. In addition, for future increased interoperability at the application level, middleware based on CORBA is expected to be used in embedded, real-time systems [3]. Moreover, connecting tactical aircraft to the GIG will introduce additional COTS middleware and new types of applications, such as agents, into the flight software [4]. Each of these additional types of middleware and applications introduce their own security issues and vulnerabilities

Although tactical datalinks provide "in transit" security at the physical level, with the introduction of layered communication protocols, a single layer security approach is not considered sufficient. Instead, a layered defense-in-depth is needed that protects the information while it is being passed over the network (“in transit”) and as it is being processed on the host computing

platform ("at rest"). The domain analysis showed that insider attacks could take place on the host computing platforms by erroneous and/or malicious applications. Thus, the domain analysis indicated that there is the potential for passive, active, and insider attacks. These attacks could result in information being delayed, corrupted, exposed, or originated from an unknown source.

EISA Threat Analysis

Figure 2 illustrates a time-critical target scenario in a network-centric battlespace on which a threat analysis was performed. In the scenario, an unmanned air vehicle (UAV), such as a Global Hawk, detects a potential threat, which it sends to the C2 platform (e.g., AWACS). As a result, the AWACS sends commands to the tactical aircraft and to satellites, which then send back additional reconnaissance data.

The threat analysis found that the tactical platform was vulnerable to the following types of information warfare attacks.

- **Spoofing** -The messages are not coming from or being received by the C2 officer responsible for the tactical aircraft or the messages are not being received by or coming from the tactical aircraft

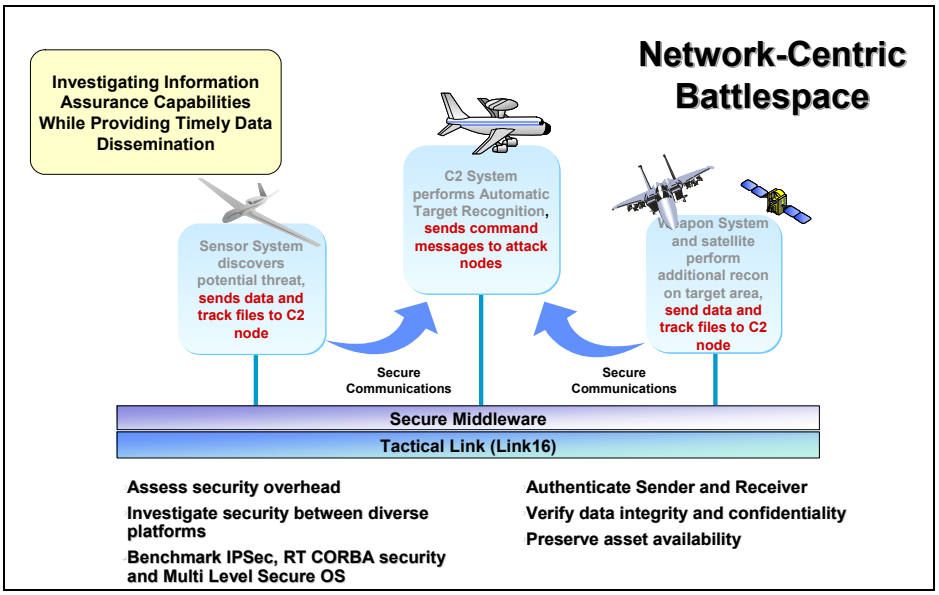

Figure 2 Time-critical Target Scenario in a Network-centric Battlespace

- **Sniffing/Traffic Analysis** - Some unauthorized platform, object, or individual is reading the transmissions or analyzing the message traffic.
- **Denial of Service**
  - **Flooding** - Extra messages are sent to the tactical aircraft, resulting in denial of service and possibly overwhelming its processing capability.
  - **Hijacking** - A required communication service is hijacked and taken down, which prevents remote applications from using that service.
- **Replay** - Messages are captured and resent to delay systems or provide them with invalid or outdated information.
- **Redirection/Tampering** - Messages are captured and sent to an unauthorized destination, while dummy messages are sent to satisfy the source and destination.

EISA Demonstrations

During the EISA program, we used the threat analysis to investigate security responses of the network and middleware software. Figure 3 shows a network-centric, layered communication architecture based on commercial specifications and standards that enables interoperability at the application level.

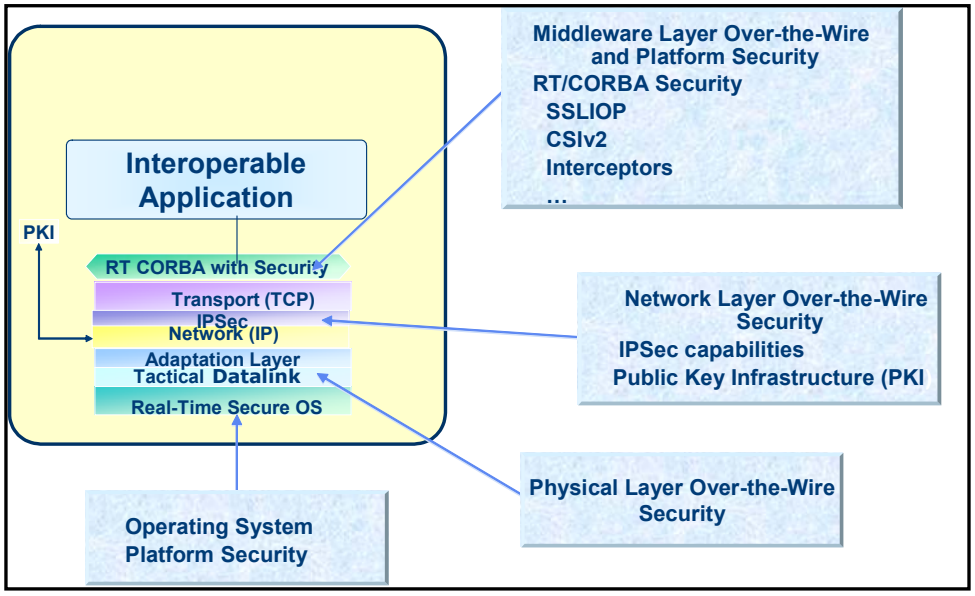

Figure 3 Network-centric Communications Architecture

In our EISA analysis, we assumed that the tactical platform would have a real-time, trusted operating system to provide security at the platform level. We also assumed that the tactical datalink would provide security at the physical communication level. At the network level, we considered attacks against TCP/IP and the effectiveness of the suite of security protocols for IP, called IPSec, in countering those attacks. At the middleware level, we considered attacks that can be made at the CORBA level and techniques provided within the CORBA specification for thwarting them.

TCP/IP Security Demonstration

The basis for secure communications on the tactical aircraft is a trusted operating system, which provides protection at the processor level, and a secure tactical data link, which provides protection at the physical interconnect protocol layer for secure, wireless, inter-platform communication. Referring to Figure 3, above the data link layer is the adaptation layer where higher level protocols interface with the data link protocol. Above the adaptation layer is the network layer, where IP sits, followed by the transport layer, where the TCP protocol resides.

Although LINK16 provides a secure, wireless communication link between C2 and tactical platforms, the TCP/IP stack sitting on top of LINK16 on each platform is vulnerable to insider attacks from other TCP/IP nodes on the platforms' internal networks. When employing these protocols, the embedded system could suffer from the same type of network attacks that assail the current Internet and desktop computers [5].

IPSec [6], the suite of security protocols for IP, sits in between the network and transport layers (i.e., between TCP and IP.) It includes two protocols, which may be applied individually or in combination.

The Authentication Header, (AH), provides packet authentication via extended IP packet header fields [8]. AH protects against replay by adding a keyed hash that prevents anyone else from retransmitting that packet. AH also prevents tampering, since the keyed hash

provides integrity assurance that packet contents have not been altered after transmission. Finally, AH prevents spoofing, SMURFing and flooding, since it offers two-way authentication that enables the client and server to both verify each other's identity

The Encapsulating Security Protocol (ESP) provides confidentiality via packet payload wrapping and encryption [9]. It creates security associations between trusted systems on the network, which maintain the ESP encryption keys. ESP prevents replay through a sequence number in the (ESP) header, which is checked to make sure the packet has not already been transmitted. ESP also prevents sniffing and traffic analysis, since encryption hides both the type and contents of valuable data.

We performed demonstrations of various TCP/IP attacks with IPv6 [10], then added IPSec, with the AH and ESP protocols enabled individually and together. A range of IP and TCP-specific attacks were explored**.** We demonstrated that IPSec encryption successfully defends against sniffing and traffic analysis. Further, we demonstrated that IPSec authentication successfully defends against spoofing, redirection, and replay. In contrast, we found that IPSec authentication was not completely successful against a denial of service attack by flooding.

RT/CORBA Security Demonstration

The Object Management Group's Common Object Request Broker Architecture (CORBA) specifies an application level protocol for distributed objects to invoke services on other objects. A CORBA ORB sits on top of the network stack to provide the application interface to remote services.

CORBA is independent of processing system, operating system, and programming language. It also provides for transparent interoperability between CORBA implementations from different vendors. Thus, it is an excellent software platform on which to build interoperable systems. Although CORBA was originally conceived for non-realtime IT applications, it is starting to be used in real-time, embedded military applications. Real-time CORBA is an extension of the original CORBA specification to provide

real-time functionality on top of CORBA's basic capabilities.

Viable RT/CORBA implementations can be obtained as COTS products or as open source products. Under EISA, we considered two such products. Objective Interface Systems' ORBexpress™ is a leading commercial RT ORB, while The Adaptive Communication Environment ORB (ACE/TAO) is a widely used and accepted open-source RT ORB originally developed at Washington University in St. Louis under the direction of Dr. Douglas Schmidt.

To provide security at the CORBA level, we looked at the various CORBA security specifications. In particular, the Secure Internet Inter-ORB Protocol (SecIIOP) specification adds security functionality to CORBA's Internet Inter-ORB Protocol (IIOP), which provides an interoperable interface to TCP/IP. The most adopted and widely distributed implementation is the Secure Sockets Layer Inter-ORB Protocol (SSLIOP).

SSLIOP uses the SSL standard for authentication and encryption over CORBA's Internet Interoperability Protocol. As with SSL, SSLIOP provides authentication and certificates for passing encryption keys. The security tags of the certificates are written in the CORBA Interface Definition Language (IDL) for the objects. Certificates are either pre-made with an already validated issuer, or a certificate issuer server can be used with OpenSSL for dynamic certificate validation. The encryption supports widely used algorithms such as RC4, DES, 3DES, and IDEA. Mild integrity is protected through Message Authentication Code (MAC) and message digests using the SHA-1 and MD5 algorithms.

SSLIOP can provide additional encryption over that provided by IPSec. In addition, it provides some measure of light authentication for protection at the object level, rather than at the principal level, where IPSec authentication functions.

To demonstrate the advantages of using SSLIOP in conjunctions with RT/CORBA over TCP/IP, we integrated the ACE/TAO ORB and SSLIOP in a scenario involving several tactical aircraft, a C2 aircraft, and an attacker.

Desktop computers were used to represent the various platforms. Tactical aircraft were represented on Windows and OpenBSD based machines. AWACS was represented on a Solaris machine and the attacks were executed from a Linux workstation.

In the demonstration, IPSec continued to provide protection at the network and transport level. In addition, SSLIOP encryption provided protection at the object level, preventing the attacker from sniffing and spoofing of the communication packets. However, SSLIOP did not prevent other types of attacks, such as flooding and message replay.

CORBA Interceptor Demonstration

The demonstrations of IPSec and SSLIOP showed good success in countering various attacks against "over the wire" transmission of information. We then went on to consider CORBA-specific attacks against information "at rest". It is generally recognized that the CORBA services such as the Naming Service and Event Service, which are especially critical to the availability of object-to-object communication across heterogeneous ORBs, are also particularly vulnerable to attack.

To provide security for the vulnerable CORBA services, we looked further into the interoperable security chapter of CORBA 3.0, in particular, the Common Secure Interoperability version 2 (CSIv2). Based on a vendor suggestion, we also looked at the applicability of the CORBA Portable Interceptor construct, which is part of the basic CORBA specification.

There are only a few vendors who implement the security features of CORBA. And there is no implementation of RT/CORBA that includes specific security features, although portions of CSIv2 are in process of being added to ACE/TAO. However, Portable Interceptors are generally available in CORBA products. This fact, combined with the currently ongoing implementation of CSIv2 in ACE/TAO, led us to experiment with using Portable Interceptors in an interoperable, CORBA-based security architecture.

One type of Portable Interceptor , called a Request Interceptor, is designed to intercept the flow of a request/reply sequence through the ORB at specific points so that services can query the request information and manipulate the service contexts that are propagated between clients and servers.

Taking the Naming Service as an example and evaluation point, the Portable Interceptors provide the hooks to halt execution at prime function calls where security should be implemented. For instance, when an object tries to bind or register in the Naming Service, a portable interceptor can be invoked, the current function halted, and the necessary information collected and passed to a security service. The authorization and access decisions are then performed according to a predefined policy.

If the policy determines that the object is allowed to bind, the binding execution is released and allowed to continue. Otherwise, an exception or warning is raised and the binding is not allowed. This provides protection for the Naming Service and other objects that use the Naming Service by not allowing invalid objects to be registered in the Naming Service for other objects to find.

The critical decision making service, preferably based on CSIv2, would only be accessible by interceptors that are automatically called when a certain type of request is received. After two objects have bound with the Naming Service, in order to communicate, a client object resolves the address of a server object by using the Naming Service. Interceptors also provide security during this step by allowing address resolution only for objects with the proper authorization. Protection can also be applied to other CORBA services by adding appropriate interceptors that hook into those other services and then invoke the same security service.

Table 2 provides a summary of the effectiveness of applying IPv6 with IPSec, CORBA's SSLIOP, and portable interceptors against various kinds of attacks.

Information Assurance in the GIG

Iin the remainder of the paper, we focus on the GIG to illustrate the kinds of capabilities that are expected to be

implemented to enable the DoD Joint Vision. As an example of enabling that vision, the multi-year, multi-phase JBI program was begun by AFRL to develop the capabilities outlined in the previously mentioned SAB report. Phase 0 of the JBI program was begun in 2001. The program is currently in Phase II, Mercury, and is expected to complete with Phase IV in 2007.

The JBI is intended to be the repository of all the Air Force's mission critical electronic data, including both historic data and real-time data feeds**.** The real-time data feeds come from intelligence and surveillance systems, theater, and national assets.

Within the JBI, C2 and tactical systems are considered nodes (IP addresses) in a Wide Area Network. A system can be a server of raw data (from onboard sensors) or a client of other information servers. Through use of the JBI, data can be accessed, searched, and manipulated to create new informationA JBI can be considered a *System of Systems* that integrates, aggregates, & distributes information in the appropriate form, at the appropriate level of detail, to users at all echelons. The JBI is based on four key concepts.

.

Table 2 Effectiveness of Defense-in-Depth Security Architecture Against Information Warfare Threats

Threat		CORBA	CORBA w/ SSLIOP	CORBA w/ Interceptors Architecture	CORBA w/ SSLIOP & Interceptors Arch.	IPv6	IPSec w/ IPv6	CORBA w/ IPSec and IPv6
Sniffing								
	Sniffing message payload		X		X		X	X
	Traffic Analysis		X		X		X	X
Spoofing								
	Spoofing packets		X		X		X	X
	Spoofing CORBA object ID			X	X	N/A	N/A	X
Denial of Service								
	Flooding						X	X
	Naming Service hijack/takedown			X	X	N/A	N/A	X
Replay								
	Replay messages						X	X
Redirect								
	Redirect network traffic						X	X
	Naming Service Hijack/Redirect			X	X	N/A	N/A	X

- **Publish, Subscribe, Query**
  - **Publish** information in the JBI
  - **Subscribe** to and receive newly published information from the JBI
  - **Query** and receive previously published information from the JBI
- **Fuselets**
  - Small, scripted Java programs that transform (filter, refine, fuse) data into knowledge
- **Force Templates**
  - Use of automated templates to reduce C2 workload
  - Information handshake between the JBI and a combat unit
- **Distributed Collaboration**
  - Distributed collaboration through shared, updateable knowledge objects

To illustrate how the time critical target scenario shown earlier in Figure 2 might be realized in a JBI, figure 4 shows a high level JBI architecture where the JBI enables information sharing between a C2 platform, a warfighter (weapon system), and a reconnaissance platform (a sensor system). Reachback to historical data at the AOC is provided by the JBI for.the C2 platform.

The four basic JBI concepts are highlighted in the figure. The Force Templates reside on the C2 platform and on the reconnaissance and tactical platforms that will access the JBI on the C2 platform. Fuselets reside on the platforms that host the JBI, both the C2 platforms and the AOC JBI servers.

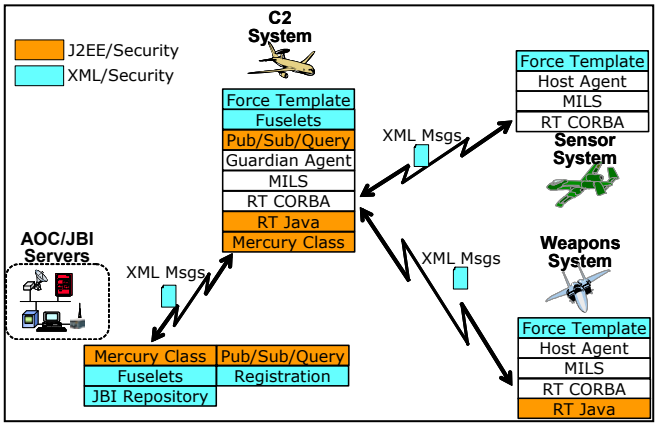

Figure 4 JBI Implementation of Time Critical Target Scenario.

As shown in the figure, messages to and from the JBI are encoded in XML and the messages are labelled using the XML security specification. Also illustrated in the figure, the JBI Pub/Sub/Query capability is implemented with the Java™ Enterprise Edition, J2EE.

The use of RT/Java™ and RT/CORBA™, as well as the host and guardian agents, also shown in the figure, are based on concepts developed under AFRL's Weapon System Open Architecture (WSOA) [3] and Insertion of Embedded Infosphere Support Technologies (IEIST) [4] programs, respectively. Multiple independent levels of security (MILS) provides a security kernel and secure middleware for real-time embedded systems. MILS is also being developed under an AFRL program of the same name.

Several IA and security concerns are highlighted in the figure. Using Java™ security, publishers and subscribers to the JBI must be authenticated and the authorization of subscribers to receive information must be verified. Fuselets, which filter and combine JBI information for clients, must be authenticated and have their authorization verified.

Under the SIRES program, we are developing a testbed for performing security experiments based on realistic mission scenarios. Shown in Figure 5, the testbed will include both workstations and embedded computer boards. We have already installed the EISA security architecture, including Ipv6, IPSec, RT/CORBA™, SSLIOP, and portable interceptors, on the testbed workstations. We have also begun installing the JBI Mercury platform, which was obtained from AFRL. The testbed will build on the EISA security architecture described in this paper, adding experiments using the security features of the JBI technologies.

Summary

EISA has demonstrated critical security features of network-level and middleware technologies for real-time embedded systems, including IPSec, CORBA SSLIOP, and CORBA Portable Interceptors. The EISA demonstrations have shown that secure communications

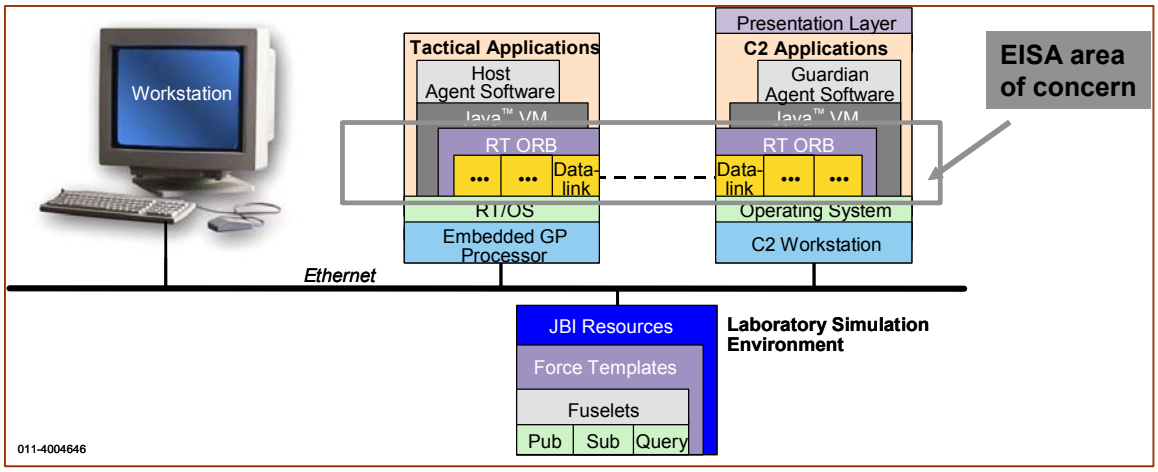

Figure 5 Conceptual SIRES Testbed Architecture

can be achieved for real-time embedded systems using commercially available technologies.

The SIRES program is extending the EISA security architecture to other middleware such as J2EE and XML, which are enablers are interoperability and the GIG. As an example of applications needed for interoperability, we are also including the Mercury JBI applications and capabilities, such as fuselets and force templates.

The information assurance technologies that were demonstrated under EISA and those that we plan to demonstrate under SIRES are key enablers for protecting tactical aircraft from Information Warfare, while providing information dominance in the air through the Global Information Grid.

References

[1] A. Miller, M. Jefferson, & J. Rogers, Global Information Grid Architecture, Mitre: The Edge, July 2001.

[2] Air Force Science Advisory Board, Information Management to Support the Warrior, SAB-TR-98-02, November, 1998.

[3] D. Corman, J. Gossett, "Weapon System Open Architecture - Using Emerging Open System Architecture Standards to Enable Innovative Techniques for Time Critical Target Prosecution", 0-7803-7034-1/01, IEEE, 2001.

[4] D. Corman, T. Herm, K. Keller, C. Satterthwaite, "Transforming Legacy Systems to Obtain Information Superiority", Command and Control Research and Technology Symposium, 2002.

[5] Oppliger, Rolf, Internet and Internet Security, Boston: Artech House, 1998.

[6] IETF RFC for IPSec AH and ESP and IPv6.

[7] S. Kent & R. Atkinson, Security Architecture for the Internet Protocol, Network Working Group, November 1998

[8] S. Kent & R. Atkinson, IP Authentication Header, Network Working Group, November 1998.

[9] S. Kent & R. Atkinson, IP Security Encapsulating Security Payload, Network Working Group, November 1998.

[10] S. Kent & R. Atkinson, Internet Protocol, Version 6 (IPv6) Specification, Network Working Group, December 1998.