Using CORBASec to Secure Distributed Aerospace Propulsion Simulations

NPSS CORBASec Test Bed
DOCsec 2001
March 28, 2001

NASA Glenn Research Center

Tammy M. Blaser
Tammy.M.Blaser@grc.nasa.gov
NPSS CORBA Sec Test Bed

Presentation Overview

• Motivation and Goal of the NPSS CORBA Sec Test Bed
• NPSS CORBA Sec Architecture *
• NPSS CORBA Wrapped Architecture *
• Schedule Phased Buildup
• Tools Current, Planned and for Future Study
• Preliminary Phase 1 of 4 Test Results and Environment
• Issues for OMG Attention
• Summary

* NPSS Dev Kit supports wrapping NPSS simulations with CORBA/CORBASec
NPSS CORBAsec Test Bed

Motivation and Goal of the NPSS CORBAsec Test Bed

• Develop a test bed using CORBAsec software and other security tools (firewalls, etc.) to secure NASA Glenn and cooperative industry partners distributed numerical propulsion simulations.

• Test bed results will drive the Numerical Propulsion System Simulation (NPSS) CORBAsec production development and deployment
  – NPSS is part of the High-Performance Computing and Communications (HPCC) Program

• The NPSS allows various aerospace companies and NASA Glenn and NASA Ames to simulate a full-scale system engine at various levels of fidelity (0D, 1D, 3D and back)
  – NPSS is built following the Object Oriented Paradigm using C++ and CORBA
  – Java developments growing rapidly (EJB based Web Servers, GUIs for testing, …)
NPSS CORBA Sec Test Bed

NPSS Production and Simulation Architecture

NPSS Production System Model

NPSS Dev. Kit supplies tools for integrating codes, accessing geometry, zooming, coupling, security.

0-D
1-D
3-D
Collaborate on Multiple Component Architecture

- Collaborative/Multiple Domain First Phase Users:
  - NASA Glenn Research Center
  - Company 1
  - Company 2
- Multiple commercial ORBs, supporting CORBA Sec ORB Interoperability, will be implemented in the final phases (scheduling details in later slides)
- Exercising use of the SSL for on wire encryption (3DES) with CORBA Sec.
- Multiple Authentication techniques implemented and planned.
- NPSS Simulation developers use a CORBA Sec enhanced NPSS API Development Kit to enable and deploy CORBA Sec.
- Configured with multiple Firewalls.
NPSS CORBASec Test Bed

Primarily CORBASeC Security-Unaware Architecture

• Primarily uses CORBASeC Security-Unaware (Enabled) *interceptor invoked* services
  – Simulation and Interpreter Interfaces with required privileges (rights) for public access
    • Security Policy Administered at Interface level with required rights for public access.
    • No methods configured for Security Policy Administration.
  – NPSS Access Control (AC) Administered based on the following combined attributes:
    • Aerospace Company or NASA Agency
    • Domain
    • Citizenship
    • Project
    • Role
    – General Users, Developers, and Restricted Users assigned privileges (rights):
      » Private access limited to General Users and Developers (General Users have read-only private access)
      » Public access granted to all User Roles (per Domain Access Policy)
      » Restricted Users limited to Public access
Plus CORBASec Security-Aware

• 100% CORBASec Security-Unaware was our design goal, but …
• NPSS AC proved runtime bound, therefore our design added a CORBASec Security-Aware application invoked “hook” and the Authorization (SecBuddy) Server was born.
• SecBuddy uses simple delegation and supports NPSS Client private access for dynamically invoked runtime bound operations.
  – via one hasPrivateAccess method call
    • Easy to update per application or security policy changes vs. changing multiple CORBASec Security-Unaware (Enabled) redundant methods
      – Old design had redundant methods get_public, get_private, set_public, set_private …
    • Low coupling with CORBASec Administration
    • Supports growth and scalability
Application Invoked AC

- Requires an Application System (NPSS Simulation, SecBuddy and Interpreter Servers) to be trusted (exercising formal application code inspections) to enforce NPSS AC decisions.
- SecBuddy Server, in a final configuration, will deploy a fault-tolerance design to make up for its dependence on Application invocation.
NPSS CORBASec Test Bed

NPSS Dynamic AC
Using CORBA Sec interceptor services and application invoked Architecture

Domain 1
Interpretor
Simulation
SecBuddy

Domain 2
Interpretor
Simulation
SecBuddy

Domain 3
Interpretor
Simulation
SecBuddy

Clients

- SecBuddy, Interpreter and Simulation boxes are NPSS CORBA Sec Servers.
- Dashed arrows are CORBA Sec security-unaware Interceptor invocations for public access.
- Solid arrows are CORBA Sec security aware Application invocations for private access.
module npssCORBA {
    interface npssObject {
        // Every npss object can hold a table of protected variables. Security Policy is not directly Administered for Interface npssObject.
        string get(in string varName);
        boolean set(in string varName,
                      in boolean isPrivate); // Many other npssObject methods exist in the production IDL
    };
    interface Interpreter: npssObject {
        // Security Policy Administered for the Interpreter Interface at the Interface level (not at parseString method) with required rights for public access.
        boolean parseString(in string cmdLine);
    };
    interface Simulation: npssObject {
        // Security Policy Administered for the Simulation Interface at the Interface level with required rights for public access.
        boolean runSim();
    };
    interface SecBuddy {
        // Security Policy Administered at Interface level with required rights. Includes one method hasPrivateAccess with required rights for private access.
        void hasPrivateAccess();
    };
};
NPSS CORBA Sec Test Bed Phase 1 Firewall Architecture

- N_ACE/Server, P_ACE/Server, and G_ACE/Server
  - Each Authenticate SecurID Clients.
- N_MSS, G_MSS, and P_MSS (Master Security Server)
  - Each Manage CORBA Sec communications and includes a LDAP Server or interface to an external LDAP Server.
  - May move MSSs into Internal networks. MSS locations TBD.
- N_Server and GP_Server
  - Each represents a Domain specific Simulation Server and a SecBuddy Server with optional Interpreter Server.
Integrated SecurID NPSS Client Authentication

- SecurID is two-factor authentication that is based on something you know (a password or PIN), and something you have (an authenticator; we use key token fobs known for their light-weight and compact size). The SecurID token generates a new, unpredictable access code every 60 seconds.
- Currently testing a SecurID workaround prototype that uses the RSA ACE/Server authentication API to integrate SecurID-based authentication with Hitachi’s TPBroker Security Service 3.4 Login API.
- The ACE/Server authentication API was provided as C libraries.
  - NASA has wrapped the SecurID authentication API methods for use with NPSS & CORBASec using C++.
- The code looks something similar to the following:

```c
NPSS prototype client {
  Do SecurID login                        // Call ACE/Server authentication APIs
  If failed, exit
  Do Security Service login              // Call Security Service login API (necessary to create Credentials)
  If failed, exit
  Continue with processing               // Processing can continue only if both authentications succeed
}
```

- After Hitachi Security Service 4.0 has implemented SecurID authentication

```c
NPSS 4.0 client {
  Do Security Service login              // Call Security Service login API now integrated with SecurID
  If failed, exit
  Continue with processing               // Processing can continue if authentication succeeds
}
```
NPSS CORBAWrapped Architecture

- NPSS is a component-based object oriented engine simulator.
- NPSS supports the use of external codes to extend a simulation to a higher fidelity and/or multidisciplinary analysis.
- Codes are CORBA wrapped to allow communication between NPSS and external codes.
- External codes use a direct CORBA wrapping scheme.
- Direct CORBA wrapping is accomplished via the easy to use NPSS Dev Kit (NPSS API for CORBA developers)
  - Transparent to the NPSS CORBA developer the API modifies the external code to become a CORBA Server conforming to the npssCORBA IDL
- The test bed results will provide the development roadmap required to add production grade CORBAsec to the NPSS Dev Kit.
NPSS CORBA Sec Test Bed

Schedule Milestones Phased Buildup

- Developed in four phases.
- Preliminary results of the NPSS CORBA Sec Test Bed first phase effort, will be available in March 2001, at that time, the NPSS software team will finalize the NPSS Dev Kit detail design using the BOA based CORBA Sec and using the POA based CORBA Server.
- The second phase will be completed in June 2001 and will support the POA CORBA Sec architecture and NPSS training events; to include a Dry Run of NPSS Dev Kit Training.
- A release of CORBA Sec will be provided for the majority of the NPSS platforms with the November NPSS Release.
- The third phase will be completed in December 2001 and will support the POA CORBA architecture and NPSS Dev Kit Training.
- The NPSS Onsite is scheduled in December 2001.
- The fourth and final phase release will support all NPSS platforms and will be provided in the February 2002 NPSS Release.
  - Due to a recent project budget cut back the final NPSS CORBA Sec release is expected to slip 4-6 months.
# NPSS CORBA Sec Test Bed

## Detailed Task Schedule

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>ORB Arch.</th>
<th>ORB</th>
<th>ORB Ver</th>
<th>O.S.</th>
<th>Qtr 3</th>
<th>Qtr 4</th>
<th>Qtr 1</th>
<th>Qtr 2</th>
<th>Qtr 3</th>
<th>Qtr 4</th>
<th>Qtr 1</th>
<th>Qtr 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NPSS CORBA Sec Test Bed Solaris Java</td>
<td>BOA</td>
<td>TPBroker Java</td>
<td>3.3.2</td>
<td>Solaris</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>NPSS CORBA Sec Test Bed Solaris Reconfigure Java and C++</td>
<td>BOA</td>
<td>TPBroker Java &amp; C++</td>
<td>3.2/3.4</td>
<td>Solaris</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>NPSS CORBA Sec Test Bed (Phase 1) Solaris Preliminary Testing Java and C++</td>
<td>BOA</td>
<td>TPBroker Java &amp; C++</td>
<td>3.2/3.4</td>
<td>Solaris</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>NPSS Onsite NPSS Dev Kit Briefing</td>
<td>BOA</td>
<td>VisiBroker</td>
<td>4.1</td>
<td>Solaris</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>DOC Sec 2001 NPSS CORBA Sec Presentation</td>
<td>BOA</td>
<td>TPBroker Java &amp; C++</td>
<td>3.2/3.4</td>
<td>Solaris</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>NPSS Training Release w/CORBASec (BOA)</td>
<td>BOA</td>
<td>TPBroker C++</td>
<td>3.4</td>
<td>Solaris</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>NPSS CORBA Server POA Solaris CR</td>
<td>POA</td>
<td>VisiBroker C++</td>
<td>4.1</td>
<td>Solaris</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>NPSS CORBA Server (POA) Dry Run Training (Phase 2)</td>
<td>POA</td>
<td>VisiBroker C++</td>
<td>4.1</td>
<td>Solaris</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>NPSS CORBA Server Port HP-UX</td>
<td>POA</td>
<td>VisiBroker C++</td>
<td>4.1</td>
<td>HP-UX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>NPSS CORBA Sec Port Solaris</td>
<td>POA</td>
<td>VisiBroker C++</td>
<td>4.1</td>
<td>Solaris</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>NPSS CORBA Server Port NT</td>
<td>POA</td>
<td>VisiBroker C++</td>
<td>4.1</td>
<td>NT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>NPSS CORBA Server Port Linux</td>
<td>POA</td>
<td>VisiBroker C++</td>
<td>4.1</td>
<td>Linux</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>NPSS CORBA Server Port Irix</td>
<td>POA</td>
<td>Mico C++</td>
<td>2.3.x</td>
<td>Irix</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>NPSS Release w/CORBA Server (POA) Unsecured Majority OS</td>
<td>POA</td>
<td>VisiBroker C++</td>
<td>4.1</td>
<td>All w/o Irix</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>NPSS CORBA Sec Merge Solaris/HP-UX/NT</td>
<td>POA</td>
<td>VisiBroker C++</td>
<td>4.1</td>
<td>Solaris/ HP-UX/NT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>NPSS Release w/CORBA Sec (POA) Majority OS</td>
<td>POA</td>
<td>VisiBroker C++</td>
<td>4.1</td>
<td>Solaris/ HP-UX/NT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>NPSS Onsite Dev Kit Training (Phase 3)</td>
<td>POA</td>
<td>VisiBroker C++</td>
<td>4.1</td>
<td>Solaris/ HP-UX/NT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>NPSS CORBA Sec Merge Linux</td>
<td>POA</td>
<td>VisiBroker C++</td>
<td>4.1</td>
<td>Linux</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>NPSS Release w/CORBA Sec (POA) Linux</td>
<td>POA</td>
<td>VisiBroker C++</td>
<td>4.1</td>
<td>Linux</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>NPSS CORBA Sec (MicoSec) Irix CR</td>
<td>POA</td>
<td>Mico C++</td>
<td>2.3.x</td>
<td>Irix</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>NPSS CORBA Sec (MicoSec) Merge Irix</td>
<td>POA</td>
<td>Mico C++</td>
<td>2.3.x</td>
<td>Irix</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>NPSS Web Security Merge</td>
<td>POA</td>
<td>VisiBroker Java …</td>
<td>4.x</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>NPSS Release w/CORBA Sec &amp; Web Security (Phase 4)</td>
<td>POA</td>
<td>Visi Java/C++ &amp; Mico</td>
<td>various</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Tools Current

- Solaris 2.6 (2.8 - later phases)
- Hitachi TPBroker Security Service (SS) for Java and C++ 3.4
- Hitachi TPBroker (VisiBroker repackaged/hardened (BOA) ORB) for Java and C++ 3.x
- LDAP iPlanet Directory Server 4.12
- Sun JDK 1.2 (1.3 – later phases)
- Sun Sparcworks 5.0 C++ compiler
- RSA ACE Server and Agent v.4.1 for Solaris
- NAI Gaunlet 5.5 Firewall (6.0 – later phase)
- Checkpoint 4.0 Firewall
NPSS CORBA Sec Test Bed

Tools Planned

• In June, Phase 2 we will
  – Upgrade to Forte 6.1 C++ compiler.
  – Upgrade to Solaris 2.8.
  – Upgrade CORBA Sec using Hitachi SS 4 to POA based on VisiBroker 4.x. ORB.

• Phase 3 we will add
  – Ports to HP/UX 11, Linux RedHat 6.2 and (NT 4 and/or Windows 2000)
    • Additional C++ compilers (HP aC++, GCC, Microsoft, …)

• Phase 4 we will add
  – Port to Irix using MICO 2.3.4 ORB
    • Irix C++ compiler
  – MICOSec for NPSS CORBA Sec Irix implementation
  – EJB/J2EE Web Security (BEA WebLogic)

• Need to add software tools to support:
  – Security Policy Generation
    • Develop detailed Security Policy Plans
    • If an intruder breaks in will our policy be?
      – Shutdown or Track the intruder?
NPSS CORBASec Test Bed

Tools Planned/Recommended

– Intrusion Detection
  • Choose a Intrusion Detection System based on detailed NPSS Security Policy Plans
  • Recommend Security Interfaces between CORBASec Non-Repudiation and Intrusion Detection Systems

– Certification
  • FIPS 140-1
    – FIPS 140-2 - when approved will replace FIPS 140-1
    – References Common Criteria
    – RSA BSAFE Crypto-C FIPS 140-1 (level 1) Certified (Integrated in VisiBroker SSL Pack)
  • Common Criteria
    – Solaris 2.8 (EAL 4) certified, but what about CORBASec?
    – CORBASec not ready, but to meet near term NPSS schedule …
    – Recommend SSL to be Common Criteria Certified. Why?
      » ANSI X.9F is in the process of embracing FIPS 140
      » IV&V good engineering assessment
    – Pursue SSL Common Criteria (EAL 3) Certification for:
      » RSA BSAFE SSL-C and SSL-J (Integrated in VisiBroker SSL Pack)
      » Baltimore Technologies KeyTools SSL (Integrated in Orbix 2000; beta now)
Tools for Future Study

- Need to study potential of using other Authentication Controls:
  - PKI Digital Certificates/Smart Cards
    - Methods for managing multiple CA endorsed digital certificates
  - Biometrics
    - Fingerprint
    - Retinal Scan
    - Iris Scan
    - Voice Recognition
    - Face Recognition
Preliminary Phase 1 of 4 Test Results

• Test Build Up Approach:
  – CORBA only Site Specific
  – CORBASec Site Specific
  – CORBASec Collaborate Network

• Completed IIOP proxy (CORBA only/non-CORBASec) tests between Site 1 DMZ (Firewall) and Internal network:
  – Tests Used Hitachi TPBroker Security Service example Client/Server s/w.
  – And NAI Gaunlet 5.5 Firewall IIOP Proxy

• Completed plug proxy CORBASec tests between Site 1 DMZ (Firewall) and Internal network:
  – Test Used SecurID and CORBASec Login workaround prototype.
  – Test Checked Out our NPSS CORBASec Prototype.
    • With SecBuddy, Interpreter and Simulation NPSS CORBASec Servers with various Clients.
  – Used NAI Gaunlet 5.5 Firewall Plug Proxy
    • SSL Proxy not used until enhancements made and Firewall Traversal Specification Baselined.
NPSS CORBASec Test Bed

Preliminary Phase 1 of 4 Test Environment

- Test bed effort requires coordination with many groups (networking, developers, system administrators, various companies, etc.)
  - Network engineers
    - Are the “keeper of the keys”
    - Do not know CORBA/CORBASec (but Network engineers are gaining elevation)
    - Understand Firewalls and VPNs
- Hitachi has proved to be a very professional company and their company policy is to actively work with the NPSS CORBASec project.
- Currently we are moving forward with our first set of collaborative (between companies) CORBASec tests.
NPSS CORBAsec Test Bed

Issues for OMG Attention

• No Standard SSL API
  – SSL API required for multi ORB vendor CORBAsec.
    • SSL Portability interfaces implemented by Hitachi Security Service as they port to both VisiBroker 4.x and Orbix 2000 ORBs.
    • SSL Interoperability required for end-to-end communication between NPSS partners.

• No Standard LDAP API
  – LDAP API will reduce risk of integrating multiple LDAP Servers (Site and CORBAsec specific).

• Need Firewall Traversal Specification Baselined
  – Forward Identity (Delegation) of Client/Server Credentials may be required for the NPSS Collaborative Project in later phases.
  – Bi-Directional GIOP
Summary

- We are integrating a production grade CORBAsec capability with our component-based object oriented engine simulator (NPSS)
  - Porting to
    - Commercial ORBs (VisiBroker 4.x and Orbix 2000)
    - C++ and Java
    - Multiple operating systems (Solaris, HP/UX, NT, Windows 2000 and Linux)
      - With dedicated MICOSec development for SGI Irix platform
- Our test bed effort is key to the safe use and success of the NPSS project in its deployment phase.