Case Study: Enhancing IBM Websphere with SELinux

Karl MacMillan
Tresys Technology

2006 SELinux Symposium
Project Overview

- Primary development targets
  - UK Government pilot program
    - demonstrate effectiveness of SELinux
    - meet security needs of UK Government
  - Create prototype IBM Websphere enhancement
    - explore adding SELinux to complex middleware
    - support wider configurations than pilot
- Timeline and status
  - Proof-of-concept / demo nearing completion
  - Initial prototype in development – complete 4/1
  - Pilot rollout planned for 5/1
WebSphere Prototype Goals

- Enhance the security of WebSphere using SELinux
  - Server process confinement and protection (sandboxing)
  - Strongly enforce the N-tiered architecture
  - Fine-grained, configurable network security at the process level
- Security configuration familiar to a typical WebSphere admin
  - Requires no SELinux knowledge
  - Eventually integrated into standard Websphere admin tools
- Support typical WebSphere functionality
  - Following best-practices from IBM hardening guides
  - Better security without limited functionality
  - Ideally support features with scalable security
    - security features increase / decrease based on configuration
Pilot Targeted Configuration

- **Client (web)**
- **fw**
- **http**
- **fw**
- **Production LAN**
  - **WAS Cluster**
    - **WAS**
    - **Deployment Manager**
- **LDAP**
- **DB2**

- **fw**

- **Data Repository**
- **Policy Development Target**
- **Non-Targeted System**
- **Network Device**

© 2006 Tresys Technology, LLC (www.tresys.com/selinux, selinux@tresys.com)
Websphere Application Server Architecture

WAS

Admin Services

Node Agent

App server

App server

Admin LAN

eth2

eth0/eth1

Production LAN

= JVM Application

= System Application

= Network Interface
Challenges

- SELinux provides strong base
  - Provides process confinement and protection
  - Can tighten the network at the process level
- Current policies are rigid, inflexible
  - Take it or leave it for each target in the policy
  - Targets get broad networking permissions
- Customization requires writing policy
  - Even to make minor adjustments, e.g., ports
  - Potentially for dozens of systems
- Administration requires policy updates
WebSphere Prototype using “razor”

Application Security Configuration File (websphere.conf)

razor application translator (rtrans)

razor core technology (rcore)

Application Packages (rpackage)

Application Policy Modules (rmod)

Deploy Packages to App Node

Existing Application Install & Configuration Tools/GUI

Hand Editing
Future Functionality

Application Security Configuration File (websphere.conf)

Razor application translator (rtrans)

Razor core technology (rcore)

Application Packages (rpredicate)

Application Policy Modules (rmod)

Packages Automatically Deployed
Benefits

- Server process confinement and protection
  - Last line of defense in case of errors / exploits
  - Protect Websphere from other services
- Provide flexible, process level network control
- Strongly enforce N-tier architecture
  - Supports the Websphere security model
- Enable server consolidation
  - Multiple application servers per system
  - Safely accessing backend data of different sensitivity
- All the benefits of SELinux
  - Specifically tailored for each deployment
  - Without detailed SELinux knowledge required
QUESTIONS?
Backup
Pilot Supported Features

- Multiple IBM HTTP Servers
- WebSphere Application Server Network Deployment
  - Single WAS cluster
  - multiple nodes
  - multiple app servers per node
- Deployment Manager
- Practically any backend services
  - DB2 (app data and session), IBM Directory Services, etc
  - Access is allowed based on IP address, port, netif
- General administrative services (Tivoli, etc)