RFP 78 Security Use Cases

Open Distribution, Draft

7 Pages

Abstract

The current set of security permission classes in OSGi does not provide enough flexibility to express security requirements in an enterprise application server environment. This paper outlines the use cases that are not supported by today's OSGi permission classes.

0 Document Information

0.1 Table of Contents

0.2 Terminology and Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in 6.1.

```
Source code is shown in this typeface.
```

0.3 Revision History

The last named individual in this history is currently responsible for this document.

All Page Within This Box

Revision	Date	Comments
Initial	March 22, 2007	Initial draft created. John Wells, BEA Systems, Inc. jwells@bea.com
0.1	April 17, 2007	Fold in comments from various reviewers John Wells, BEA Systems, Inc. jwells@bea.com
0.3	May 15, 2007	F2F Review Comments John Wells, BEA Systems, Inc. jwells@bea.com

1 Introduction

There are many products today being written on top of an OSGi framework that work like an application server, in that third-party bundles are deployed along-side of trusted system code. Both the system bundles and the third party bundles share things such as the package and service space. The current security model of OSGi is rich and allows for very flexible handling of groups of bundles based on conditions.

However the set of permissions for OSGi related items such as packages and services do not allow for certain use cases that become important in an application server environment. It is important when running in such an environment that both the system code and the user code be given certain guarantees about the security of the system in order to have trust in adopting the service platform.

2 Application Domain

The application domain is OSGi server applications where third party bundles may be deployed alongside trusted bundles from the application server provider. The ability to split the set of bundles into categories like these is already well defined in the Conditional Permission Admin Service.

2.1 Terminology + Abbreviations

- Application Server: In this paper an application server refers to a running OSGi system that has trusted system bundles written by the application server vendor running alongside third-party user bundles that are provided by the end customer (i.e., not the application server vendor). It does not imply a JEE, .NET or database server, but simply a server whose job it is to host other services and provide them a consistent set of system services upon which they can rely.
- Conditional Permission Service: This is the service defined in Chapter 9 of the OSGi Service Platform Core Specification (R4). It defines conditions, permissions and how security is applied in the R4 OSGi platform.

All Page Within This Box

3 Problem Description

The existing permission classes in OSGi do not have the flexibility needed to handle certain use cases. This includes but is not limited to

- org.osgi.framework.PackagePermission
- org.osgi.framework.ServicePermission
- org.osgi.service.event.TopicPermission
- org.osgi.service.wireadmin.WirePermission
- org.osgi.service.cm.ConfigurationPermission

Providing these capabilities on the OSGi platform will facilitate the adoption of OSGi into enterprise application server environments.

4 Use Cases

This is not an exhaustive list of use cases that should be supported by any RFC in response to this proposal, but should instead be treated as representative of typical scenarios.

These use cases are all enterprise systems where there are some bundles running from trusted sources (system bundles) and other bundles that are dynamically installed by a deployment subsystem (user bundles). The ability to split bundles in this manner is already well defined in the Conditional Permission Admin specification. The user bundles may not be signed and most of them will not have an OSGI-INF/permissions.perm resource. In this environment it is assumed that there is a system bundle that is granting permissions to the user bundles as they are installed.

Note that while the use cases below break the bundles into system and user bundles for simplicity, any arbitrary breakdown of bundle groups or domains as defined by the conditional permission service conditions MUST be supported.

4.1 Deployed bundles should not be able to offer a specific service

This is an enterprise system as described above.

Suppose there is a service called “org.acme.security.AuthorizationService” that only the system bundles should be able to register. The user bundles are still allowed to register services, just not the org.acme.security.AuthorizationService. It is not known at the time of deployment the set of services that the user bundle might offer.

Using the current conditional permission service and org.osg.framework.ServicePermission it is not possible to express this requirement. The ServicePermission allows administrators to grant permissions to offer certain services but not to deny them.

All Page Within This Box

To accomplish this today would require the system bundle that is granting permissions to list all of the services that the user bundle could offer, without knowing beforehand what services the bundle might offer (and **not** include in that list the service it should not offer). What makes this worse is that some user bundles may not know themselves what services they are going to offer. For example, the user bundle might be one that offers proxies of services for other bundles with some extra quality of service for the customer, such as metering or load balancing.

4.2 Deployed bundles should not be able to get specific services

This is in the same system as above, except that this time the system bundles want to restrict the user bundles from getting a specific service

Suppose there is a service named “org.acme.security.UserAdministratorService” that allows management consoles to add and remove users from the user database. The trusted management console bundle needs to use this service in order to administer users. However, the user bundles in the system should not be able to get access to this service.

Using the current conditional permission service and org.osgi.framework.ServicePermission it is not possible to say “only system bundles may GET this service”. The user bundles should be able to get any other service, but not “org.acme.security.UserAdministratorService”. There are services coming and going dynamically all the time (including some from deployed bundles), so having an exhaustive list of services that the user bundles can access is not feasible. The ServicePermission allows the system bundle granting permissions to grant permission to **get** services but not to **deny** access to particular services.

4.3 Deployed bundles should not be able to get specific packages

This is in the same system as above, except that this time the system bundles want to restrict the user bundles from importing or exporting a specific package.

Suppose there is a package named “org.acme.security.secret.sauce” that contains special purpose classes that system bundles can see but deployed bundles should not. Furthermore, in order to be sure that system bundles do not mistakenly resolve to a package of that name offered by a user bundle the ability for deployed bundles to export that package also needs to be restricted.

Using the current conditional permission service and org.osgi.framework.PackagePermission it is not possible to say “only system bundles may IMPORT or EXPORT this package”. The user bundles should be able to export and import other packages, but not “org.acme.security.secret.sauce”. There are packages coming and going dynamically all the time (including some from user bundles), so having an exhaustive list of packages that the deployed bundles can access is not feasible. The PackagePermission allows the system bundle granting permissions to **grant** permission to import and export packages but not to **deny** that ability.

5 Requirements

All Page Within This Box

- The solution MUST enable restricting certain bundles from offering a service, without having to list all of the services that bundle might offer.
- The solution MUST enable restricting certain bundles from getting a reference to a service, without having to list all of the other services that bundle might want access to.
- The solution MUST enable restricting certain bundles from importing or exporting a package, without having to list all of the packages a bundle might import or export.
- The solution MAY allow for a general Permission denial model
- The solution MUST NOT alter the bundle programming model
- The solution MAY boost performance by allowing for Permission implies decisions to be cached
- Any RFC written MUST support all existing OSGi execution environments

6 Document Support

6.1 References

[1]. Bradner, S., Key words for use in RFCs to Indicate Requirement Levels, RFC2119, March 1997.

[2]. Software Requirements & Specifications. Michael Jackson. ISBN 0-201-87712-0

6.2 Author's Address

Name	John Wells
Company	BEA Systems, Inc.
Address	140 Allen Road, Liberty Corner NJ 07938
Voice	+1 908 580 3127
e-mail	jwells@bea.com

6.3 End of Document