Perimeter Service Router

Context

External applications require access to one or more Web services that are deployed within a private network. Access to the Web services and resources in the private network is restricted to authenticated users. External applications should not have access to resources used by the Web services in the private network.

Problem

How do you make Web services in a private network available to external applications without exposing resources in the private network?

Forces

Any of the following conditions justifies using the solution described in this pattern:

- **Internal Web services and dependent resources may be targeted by attackers who are external to the network**. The organization must protect Web services on the internal network, so that any attacks do not affect the internal Web services or dependent resources.
- **Attackers can gain information about the internal network, and use it to compromise the network**. The organization must not reveal information about the internal network infrastructure that can be useful to attackers.

The following condition is an additional reason to use the solution:

- **External clients need reliable access to fixed service endpoints**. The location of a Web service's internal implementation may need to change dynamically to cater for the availability of dependent resources, or to cater for maintenance and batch processing windows. External clients should be unaffected by these changes.

Solution

Design a Web service intermediary that acts as a perimeter service router. The perimeter service router provides an external interface on the perimeter network for internal Web services. It accepts messages from external applications and routes them to the appropriate Web service on the private network.

Participants

Using the Perimeter Service Router pattern involves the following participants:

- **External application**. An application located outside of the private network that needs to access the Web services in a private network.
- **Perimeter service router**. The perimeter service router is a Web service that provides access to Web services in the private network.
- **Service**. One or more Web services that are accessed by the perimeter service router.

Figure 6.1 shows a perimeter service router accepting requests from a client and routing them to other services.

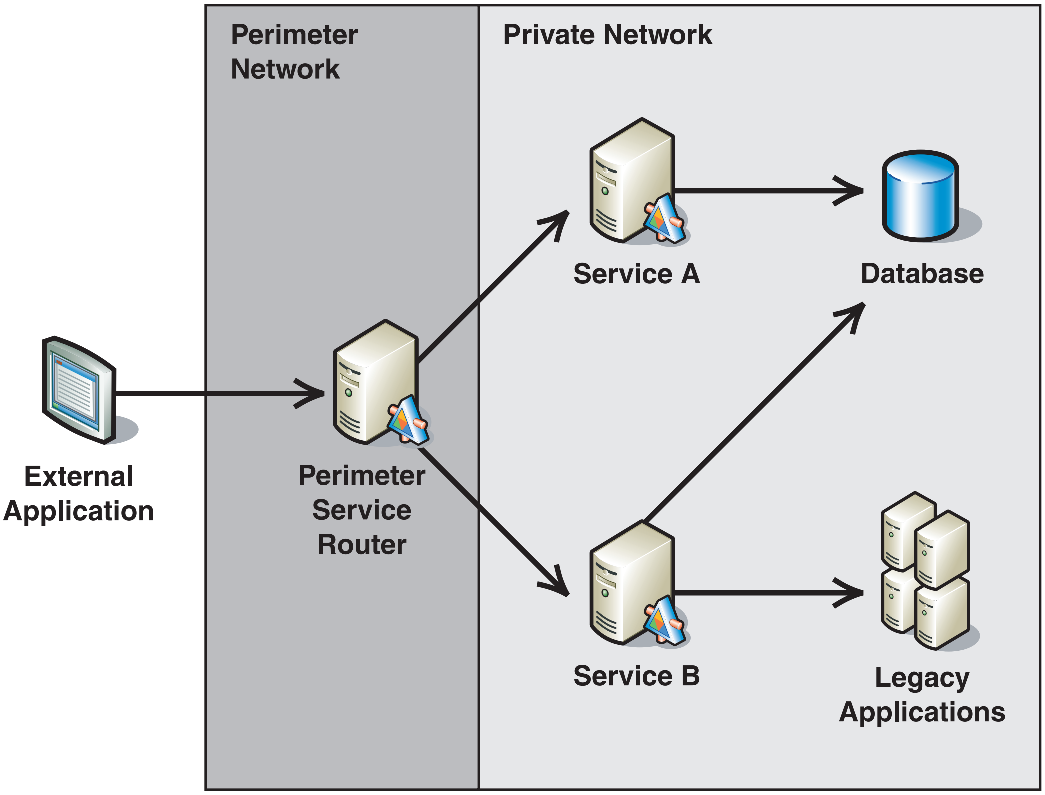

Figure 6.1
A perimeter service router on the perimeter network

The perimeter service router provides an entry point that external applications use to access the functionality exposed by internal services. The perimeter service router is typically deployed in a perimeter network (also known as DMZ or demilitarized zone), which has access to resources in the private network through a firewall. A perimeter service router operates at the application layer, and is intended to work in conjunction with existing firewall technologies and not to replace them.

Process

The following diagram illustrates the functionality of the perimeter service router.

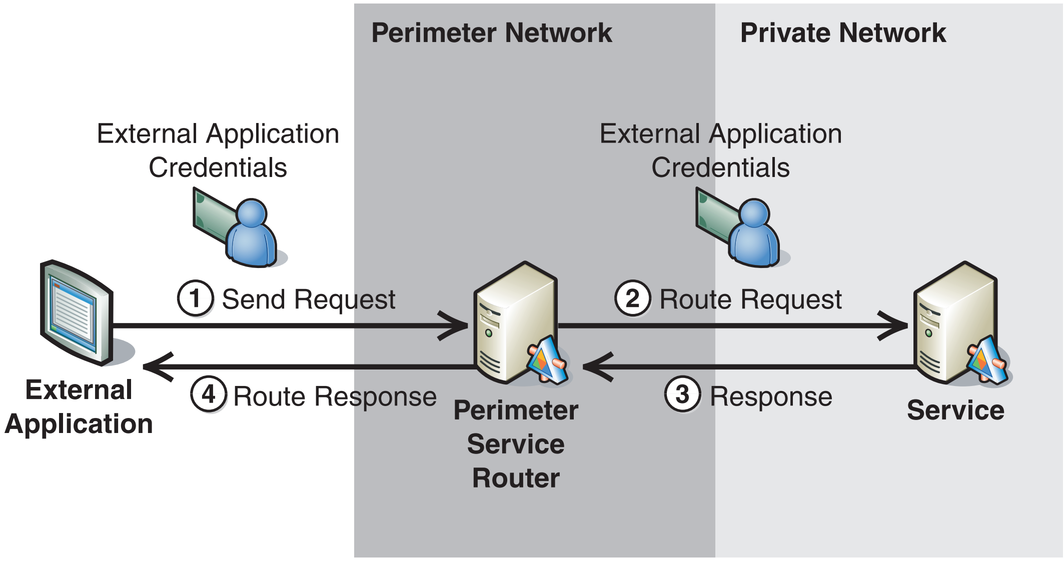

Figure 6.2
The functionality of the perimeter service router

As illustrated in Figure 6.2, the functionality of the perimeter service router is described in the following steps:

1. **The external application sends a request message**. The request message is addressed to the service's external interface on the perimeter service router. The perimeter service router typically "hides" the internal endpoint address by accepting requests through an external endpoint address that is exposed to external applications.
2. **The perimeter service router forwards the request message to the service**. The message is forwarded to the appropriate endpoint address. If the perimeter service router provides an external interface for multiple services on the private network, it will route the request to the appropriate service request based on the specific address where the request was sent.
3. **The service sends a response**. The service performs any security checks, such as authentication, and then processes the request. Based on the contract between the external application and the service, the service may send a response back to the external application.
4. **The perimeter service router forwards the response to the external application**. If the server sends a response in Step 3, the perimeter service router forwards the response to the external application.

Note: The basic perimeter service router described previously does not perform security functions as an intermediary such as authentication, replay detection or message validation. For more information about the security functions performed on a perimeter service router, see the "Benefits" section.

Example

Northwind Traders is a manufacturer that has created a suite of Web services that provide the ability to view and manage their inventory. Currently these services are only accessible to clients through a Web application provided by Northwind Traders. Many of Northwind's clients are retailers that also provide applications for their customers to order products online. When the retail customers order a Northwind Trader's product it is not possible to determine if that product is available prior to making the order. As a result, Northwind's clients would like direct access to the Web services that provide inventory information.

Instead of providing direct access to the inventory services, Northwind has decided to implement a perimeter service router that external clients can access. External clients can now incorporate calls to the perimeter service router directly into their applications to provide inventory information to their customers.

Resulting Context

This section describes some of the more significant benefits, liabilities, and security considerations of using this pattern.

Note: The information in this section is not intended to be comprehensive. However, it does discuss many of the issues that are most commonly encountered for this pattern.

Benefits

The benefits of using the Perimeter Service Router pattern include the following:

- Security can be maintained at the perimeter service router, which provides an extra layer of security to protect the Web services.
- Servers that host internal Web services can be taken offline for maintenance without affecting the external interface. This can be accomplished by configuring the perimeter service router to start routing messages to a backup server while the maintenance is being performed.
- The perimeter service router represents a single point of entry for external clients. This allows it to be extended to support additional operations that external clients require. These requirements could include:
  - **Protocol Transition**. External clients can be authenticated with different mechanisms, such as X.509 certificates, or custom authentication that is validated against a database. After the external client has been authenticated, it can be transitioned into an internal protocol, such as the Kerberos version 5 protocol to access internal Web services.

- **Message Validation**. Request messages from external clients can be validated to make sure that they do not contain malicious content prior to sending them to an internal service. Message signatures can also be validated to detect tampering.
- **Exception Shielding**. Detailed error messages that are returned by internal services can be filtered or modified prior to sending responses back to external clients.
- **Replay Detection**. The perimeter service router can keep a cache of requests and reject any duplicate requests that are sent to the interface.
- **Message Transformation**. Request messages received from clients can be transformed into a structure that internal Web services require. This provides the ability to modify internal interfaces without affecting external interfaces. It is also possible to support several structures from external clients that can be mapped into an internal structure.
- **Auditing**. Activities may need to be attributed to a specific user or organization for accounting or security auditing purposes.

Note: In some cases, you need to provide some or all of these additional requirements for internal clients as well. In these cases, you need to place the logic that provides these functions on the internal network, or ensure that the internal clients also pass through the perimeter service router.

Liabilities

The liabilities associated with the Perimeter Service Router pattern include the following:

- Many platforms make exposing the application functionality simple. However, this can lead to a poor decision in terms of granularity. If the service interface is overly fine-grained, you can end up making too many calls to perform a specific action. You need to design your service interfaces to be appropriate for network or out-of-process communication.
- Each additional service interface that a service provides increases the amount of work required to make changes to the functionality that is exposed by a service.
- The Perimeter Service Router pattern adds complexity and performance overhead that may not be justified for very simple service-oriented applications.
- The perimeter service router may become a bottleneck when routing large numbers of messages. To avoid this problem, the perimeter service router should be designed with good performance as a high priority.

Security Considerations

Security considerations associated with the Perimeter Service Router pattern include the following:

- The perimeter service router is often the only point of entry to the internal network for external clients. This can make it a prime target for attackers. To guard against an attack, you must harden the platform on which the perimeter service router is deployed.
- Although the perimeter service router can provide an extra layer of security between external clients and internal Web services on a private network, you should still ensure that you design secure Web services on the internal network. You should also ensure that communications between the perimeter service router and internal Web services are secured.

Related Patterns

The following child pattern is related to the Perimeter Service Router pattern:

- **Implementing Perimeter Service Router in WSE 3.0**. This pattern provides steps and recommendations to implement a perimeter service router in WSE 3.0. It also discusses extensibility points in the **SoapHttpRouter** class in WSE 3.0 that you can use to address advanced scenarios, such as validation and dynamic routing

Implementing Perimeter Service Router in WSE 3.0

Context

You are exposing Web services deployed in a private network to external applications. Access to the Web services and resources in the private network is restricted to authenticated users. Any applications external to the private network must use a perimeter service router to access the Web services and resources deployed in the private network.

Objectives

The objectives of this pattern are to:

- Use a perimeter service router to provide an additional layer of security for services exposed to external clients.
- Allow the perimeter service router to route information to internal Web services based on a location contained within a configuration file.
- Demonstrate how to implement a perimeter service router using the WSE 3.0 **SoapHttpRouter** class.
- Discuss extensibility points in the **SoapHttpRouter** class in WSE 3.0 that you can use to address advanced scenarios, such as validation and dynamic routing.