

IBM WebSphere Web Multi-Platform Configuration

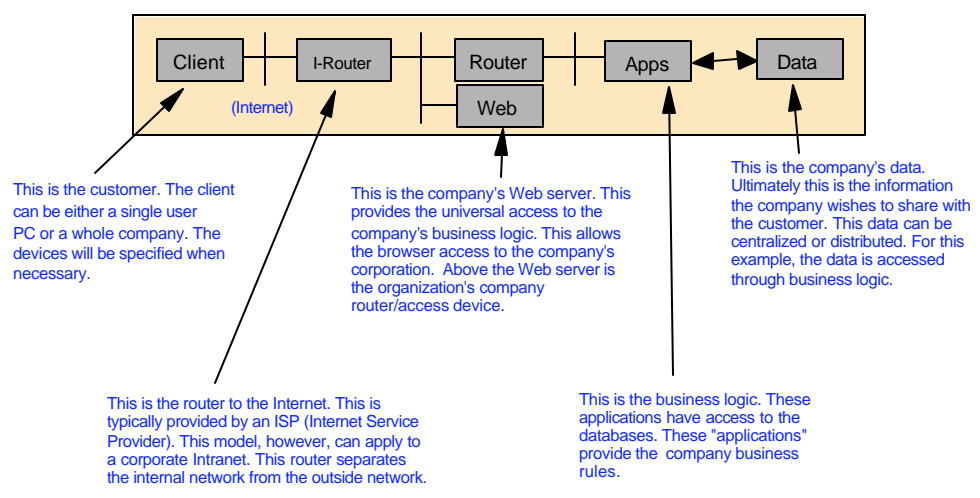
Firewalls/OSE Remote



Overview



The Basic "No Security" End-to-End Model



What is a firewall?



- A combination of specialized hardware and software that is designed to keep unauthorized users from accessing the information that is within a networked computer system



- To understand how a firewall works, imagine that your network is a building to which you want to control access. Your building has a lobby as the only entry point. In this lobby, you have receptionist to welcome visitors, security guards to watch visitors, video cameras to record visitor actions, and badge readers to authenticate visitors who enter the building.
- These measures may work well to control access to your building. But, if an unauthorized person succeeds in entering your building, you have no way to protect the building against this intruder's actions. However, if you monitor the intruder's movements you have a chance to detect any suspicious activity from the intruder.
- When you define your firewall strategy, you may think it is sufficient to prohibit everything that presents a risk for the organization and allow everything else. However, because computer criminals constantly create new attack methods, you must anticipate ways to prevent these attacks. As in the example of the building, you also may need to monitor for signs that, somehow, someone has breached your defenses.
- In the case of a firewall, your best strategy is to permit only those applications that you have tested and have confidence in. If you follow this strategy, you must exhaustively define the list of services you must run on your firewall. You can characterize each service by the direction of the connection (from inside to outside, or outside to inside). You should also list the users that you will authorize to use each service and the machines that can issue a connection for it.

Different Firewall Vendors Offer Different Services

- Packet filtering
- Application filtering
- Proxy server
- Network Address Translation (NAT)
- Virtual Private Network (VPN)



- A proxy server is a server that provides access to files from other servers by retrieving them either from its local cache or from the remote server.
- VPN allows secure communication across the Internet by setting up an encrypted link between two computers.

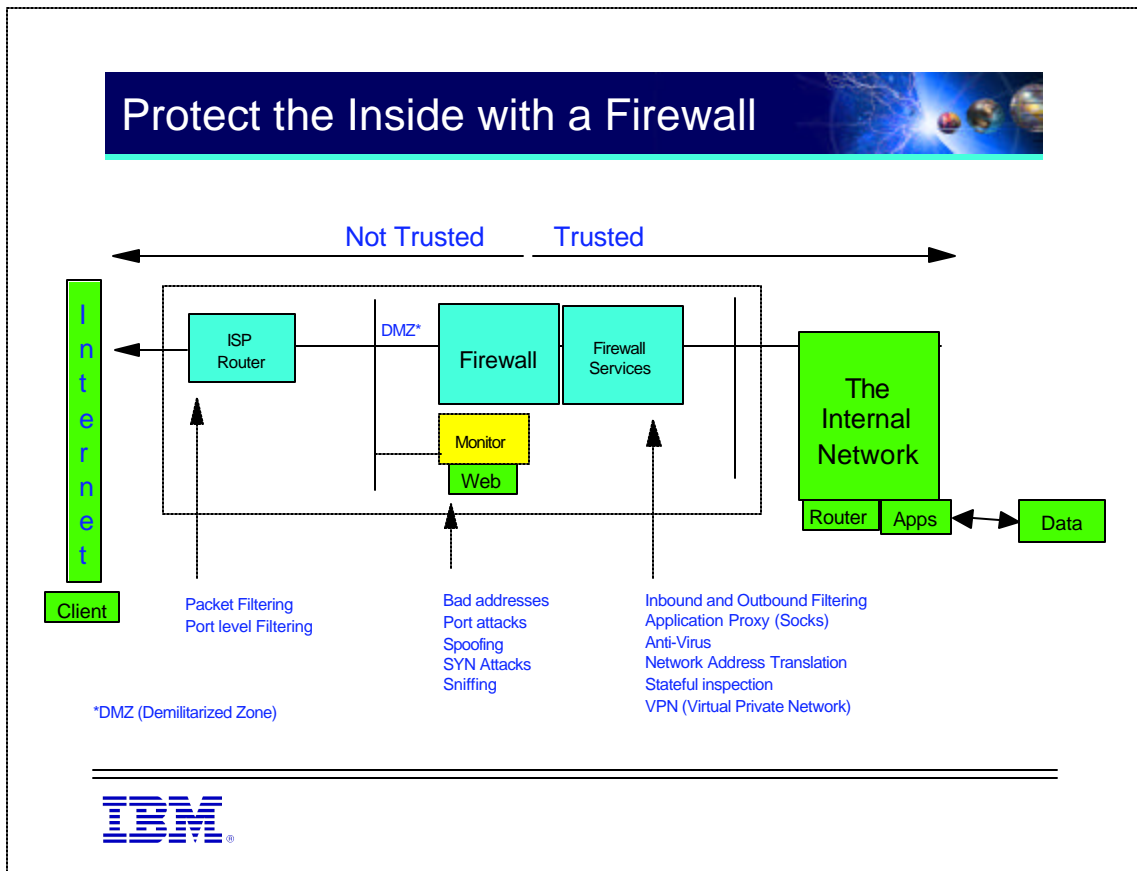
Demilitarized Zone (DMZ)



- Many networks use multiple firewalls to protect their network.
- A DMZ is a firewall architecture that employs two routers to filter and transfer information between an organization's internal network and the Internet.



Protect the Inside with a Firewall



- In order for a hacker to compromise the data on the internal network, he must breach two firewalls - the ISP Router and the DMZ firewall.
- Web servers frequently run in the DMZ while business logic applications (EJBs, some servlets, and so forth) run in the internal network.

DMZ Solutions



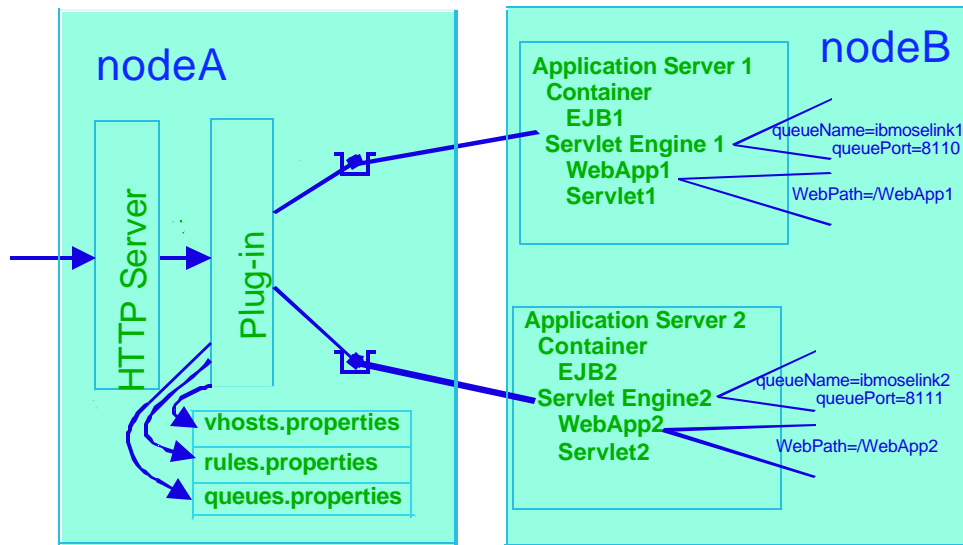
- OSE Remote
- Servlet Redirector
- HTTP Reverse Proxy



OSE Remote



OSE Remote Diagram



- hosts.properties
 - localhost=default_host
 - 127.0.0.1=default_host
 - wasedu=default_host
- rules.properties
 - default_host/WebApp1/Servlet1=ibmoselink1
 - default_host/WebApp2/Servlet2=ibmoselink2
- queues.properties
 - ose.srvgrp=ibmoselink1,ibmoselink2
 - ose.srvgrp.ibmoselink1.clonescount=1
 - ose.srvgrp.ibmoselink1.clone1.port=8110
 - ose.srvgrp.ibmoselink1.clone1.host=nodeB
 - ose.srvgrp.ibmoselink1.clone1.type=remote

OSE Remote Advantages



- Web server only needs very lightweight WebSphere install
- Very little performance hit compared to Web server and application server on a single box



OSE Remote Disadvantages



- Does not SSL encrypt the communications between the Web server node and the application server
- Requires one firewall hole for each servlet engine and one firewall hole for WebSphere security
- Supported only by product Versions 3.021 and higher
- Requires manual configuration at initial setup and any time the topology changes
- Administrative server is the single point of failure for secured Web server

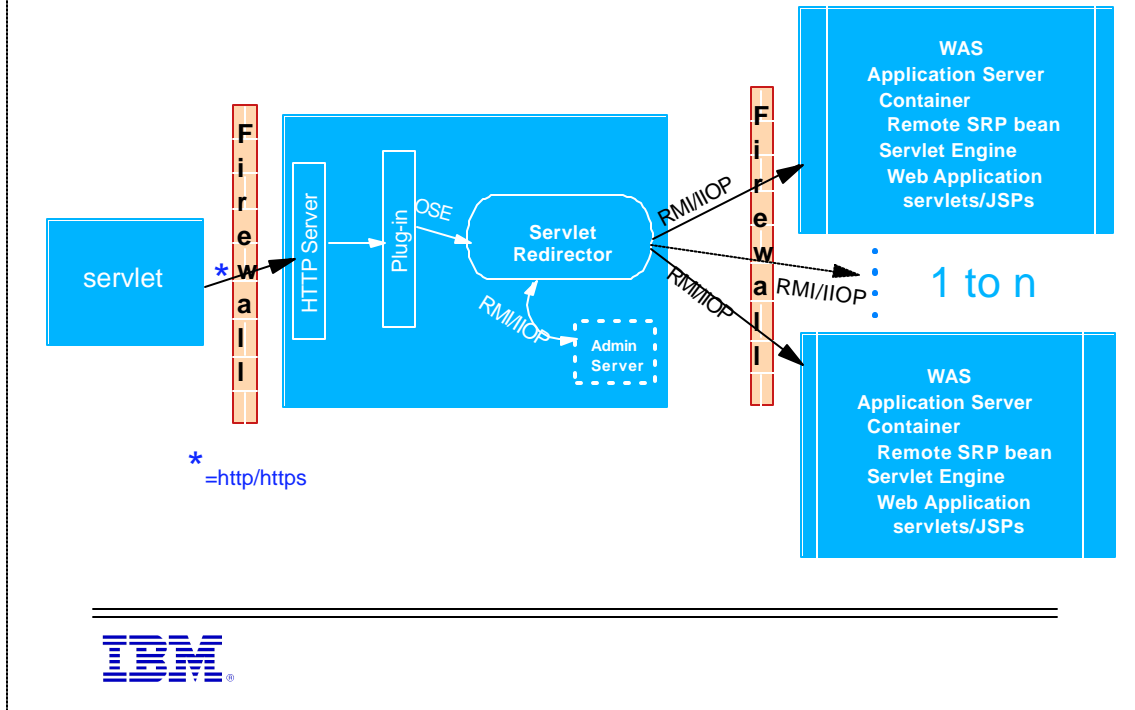


- HTTPS can still be used between the client and the Web server (It is the only communication between the HTTP server and WebSphere that is unencrypted)

Servlet Redirector



Overview of Servlet Redirector



- The servlet redirector acts as a client to the Remote SRP bean, which is shipped with WebSphere
- The servlet redirector queries the administrative server to determine which application server to send the servlet request.
- Servlet redirector forwards the request to the application server's RemoteSRP bean.
- RemoteSRP EJB forwards the request to the in-process servlet engine.
- In the case of multiple close available to handle the servlet request, the servlet redirector uses EJB workload management to route requests.
- Can be configured in admin server/admin agent or as a standalone process.

Servlet Redirector Advantages



- WebSphere security enables SSL encryption between Web server and application server



Servlet Redirector Disadvantages



- Requires 3+n firewall holes for n application servers
- Does not work through NAT firewall
- Standalone or thin servlet redirector does not support WebSphere security
- Administrative server redirector requires database access through firewall
- Administrative server becomes a single point of failure in both unsecure and secure environments
- Performs significantly slower than OSE-Remote

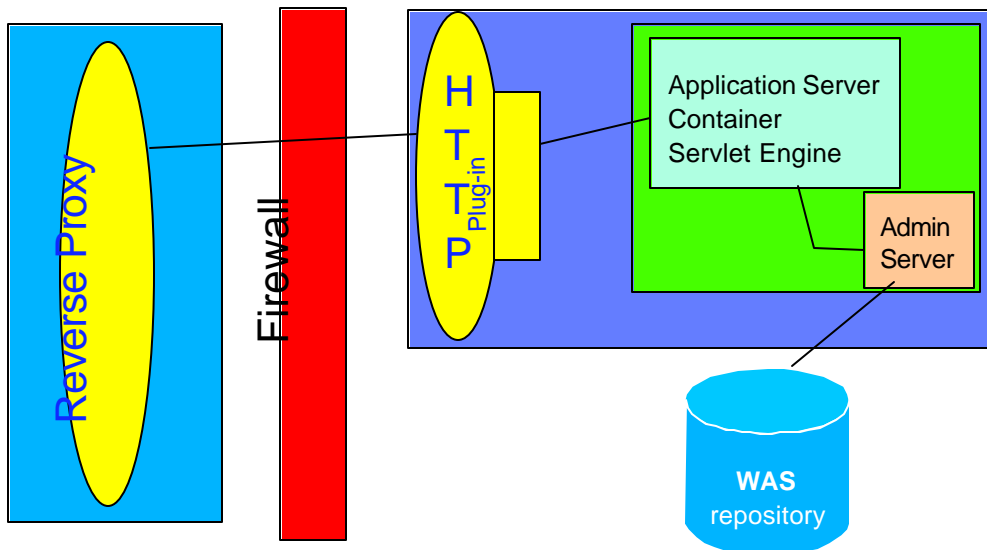


IBM WebSphere Web Multi-Platform Configuration

HTTP Reverse Proxy



Reverse Proxy Diagram



- HTTP reverse proxy leverages third part software in the DMZ to redirect HTTP requests to an HTTP server in the internal network.
- In this configuration, a Reverse Proxy that resides in the DMZ listens at port 80 for requests that have a certain format. It then forwards those requests to a HTTP Server that resides on the same machine as WebSphere. The requests are then fulfilled and passed back through Reverse Proxy to the client, hiding the originating Web server.

Reverse Proxy Advantages



- The basic Reverse Proxy configuration is well known and tested in the industry.
- WebSphere configuration is unaffected by reverse proxy



Reverse Proxy Disadvantages



- ▣ May requires more hardware
- ▣ Requires more software
- ▣ Reverse Proxy server supported by third party vendor, not WebSphere



RMI/IIOP



RMI/IOP Through Firewalls Disadvantages

- RMI over IOP picks a random port from a set of ports for listening.
- This port may change each time the process is restarted.
- To allow RMI over IOP through the firewall, a large range of ports needs to be opened.
- This causes a security hole, many companies and firewall administrators find this unacceptable.



RMI/IIOP Through Firewalls Workarounds



- All calls to EJBs use RMI/IIOP (this includes the servlet redirector)
- WebSphere created command line arguments for telling a RMI over IIOP process to listen to a specific port.





- Without WebSphere Security
 - ▲ Each application server and the administrative server has one random port
 - ▲ "Pin" these ports by specifying:
Dcom.ibm.CORBA.ListenerPort=XXXX
where xxxx is a valid IP Port on the machine



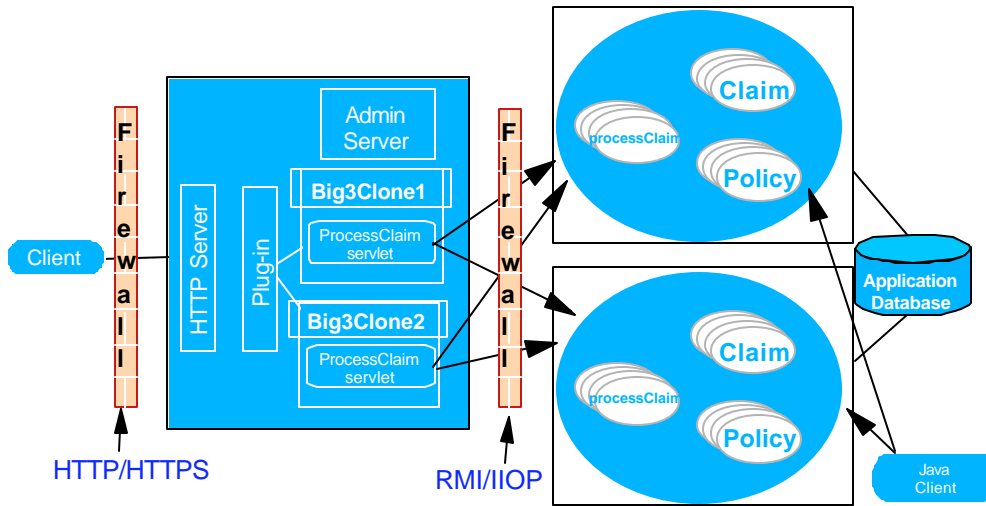
RMI/IIOP Through Firewalls Workarounds



- With WebSphere Security
 - ▲ Each application server and the administrative server also have a random port for SSL
 - "Pin" this port by specifying:
Dcom.ibm.CORBA.SSLPort=yyy
where yyy is a valid IP Port on the machine
 - ▲ Location Service Daemon has a random port for SSL communication
 - "Pin" this port by specifying:
Dcom.ibm.CORBA.LSDSSLPort=zzzz
where zzzz is a valid IP Port on the machine

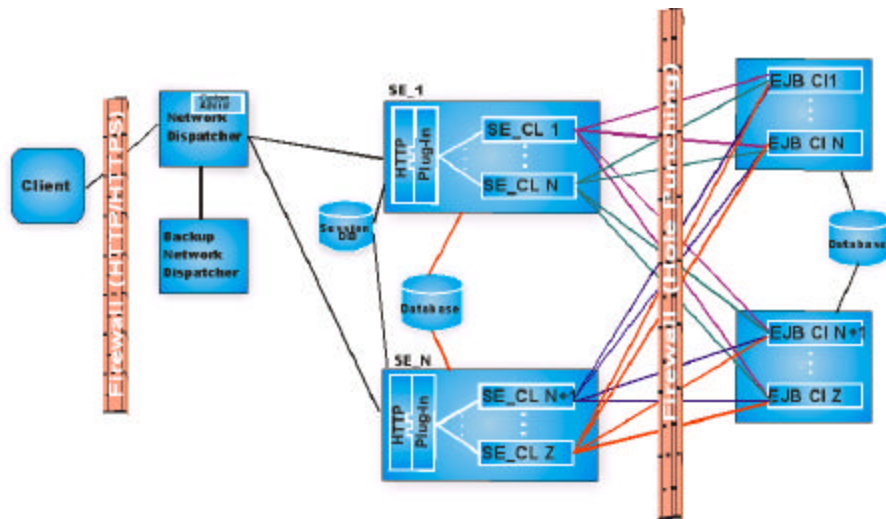


EJB Workload Management



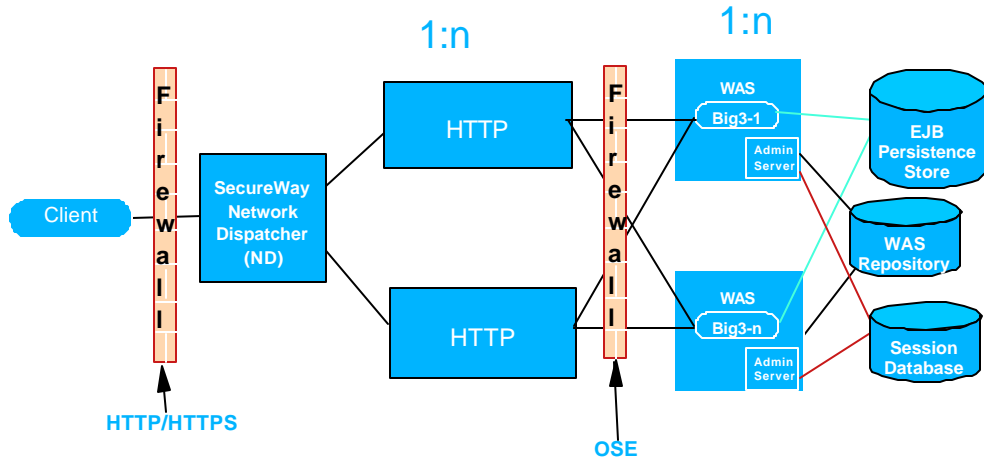
- EJBs are secure behind the firewall
- Each container holding EJBs may hold 1:n clones of the EJBs

TCP/IP Spraying, Clustering, EJB WLM



- This configuration has Network Dispatcher, with failover, running a custom advisor. The Dispatcher is routing to n instances of WebSphere. Each WebSphere instance is running clones that contain a servlet engine. Each servlet engine is running HTML, servlets, and JSPs (Servlet Clustering). Each servlet engine is communicating through a firewall to Z WebSphere instances, each running Z clones containing WLM EJBs. Sessions, if they are used, are persisted to the database.

Servlet Clustering with OSE Remote



Summary Table



Benefits of each DMZ Configuration

Configuration	Compatible with product security	Does not require database access from DMZ	Supports NAT
Thick servlet redirector	X		
Thin servlet redirector		X	
Admin-Agent with servlet redirector	X	X	
OSE remote	X	X	X
Reverse proxy	X	X	X



- **Compatible with product security**

- IBM WebSphere Application Server security protects applications and their components by using authorization and authentication policies. Configuration options are compatible with product security because they do not necessitate alternative security solutions.

- **Does not require database access from DMZ**

- A DMZ configuration protects application logic and data by creating a Demilitarized Zone between the Web site and the database that is holding these valuable resources.
- Configuration options that keep databases, and servers from directly accessing databases are more desirable. Because a WebSphere administrative server needs access to database information, it is often not a viable solution to run an administrative server in the DMZ.

- **Supports Network Address Translation (NAT)**

- A firewall product that runs NAT receives packets for one IP address, and translates the packet to send it to a second IP address.
- In environments with firewalls employing NAT, it is best to avoid configurations that involve RMI/IIOP because the IP addresses are embedded in the body of the IP packet. These addresses are not translated and make the packet useless.

Benefits of each DMZ Configuration

Configuration	Minimum required number of firewall holes	Encrypted Web server/application server communication	Supported WebSphere Level.
Thick servlet redirector	3 + 1 per application server	X	3.0x
Thin servlet redirector	3 + 1 per application server	X	3.0x
Admin-Agent with servlet redirector	3 + 1 per application server	X	3.02.1
OSE remote	1 + 1 per application server		3.02.1
Reverse proxy	1	not applicable	not applicable



- **Minimum required number of firewall holes**
 - Configurations that minimize the number of firewall ports are desirable because each additional hole leaves the firewall more vulnerable to attackers.
- **Does not require DMZ protocol switch**
 - The Web server sends HTTP requests to application servers that are behind firewalls. It is a common practice to open ports in the firewall to allow the requests through.
 - Configurations that require switching to another protocol (such as IIOP), and opening firewall ports that correspond to the protocol, are less desirable. They are often more complex to set up, and protocol switching overhead can impact performance.
- **Potential single point of failure**
 - This condition exists if the processes that are inside of the DMZ require linking to another running process (usually outside the DMZ) in order to function. Thus if one machine or process goes down, it takes another process down with it. Various failover configurations can minimize down time and help to prevent a failure, but these configurations usually require additional resources.

IBM WEBSHERE WORKSHOP - LAB EXERCISE

Configuring and Using Remote OSE

What This Exercise is About

This lab will take you through the process of configuring the Remote OSE (Open Servlet Engine) and how to use the Remote OSE link once it has been established.

User Requirement

This lab requires that you have deployed the Big3 application and created a model and two clones of this application on the SUN machine.

What You Should Be Able to Do

After you have completed this lab exercise, you should be able to:

- Configure a WebSphere servlet engine to accept Remote OSE requests.
- Configure an HTTP server to send Remote OSE requests.

Introduction

The Open Servlet Engine protocol (OSE) is an IBM proprietary protocol that is used for communications between the WebSphere Web server plugin (mainly written in C) and WebSphere, written in Java.

The underlying communication mechanism for OSE can be one of three different connection types:

- Local pipes - require that the Web Server and the WebSphere reside on the same machine. For this reason, Remote OSE requires the use of sockets.
- Sockets
- A pure Java implementation for sockets - used only for debugging.

For this lab, the Web server (Windows NT) machine has the following installed:

- Web server
- WebSphere plugin for the Web server
- Production Application Server

The application server machine (SUN) has the following installed:

- The full IBM WebSphere Application Server product
- Administrative server
- Application servers

Exercise Instructions

For this lab we will use the NT machine as the HTTP server and the SUN machine as the WebSphere machine. Estimated time: 45 minutes

Part One: Configuring remote OSE for multiple Web servers and application servers

- __1. The first step is to ensure that the Big3ServletEngine on the SUN machine is configured to use OSE INET Sockets.
- __ A. Start the administrative server and console.
 - __ B. Expand the *Big3Master* in the **Topology** tree view.
 - __ C. Locate and click *Big3ServletEngine*.



- __ D. Click the **Advanced** tab.
- __ E. Click the **settings** button to access the plugin configuration settings.
- __ F. Select **INET Sockets** for the *Transport Type*.



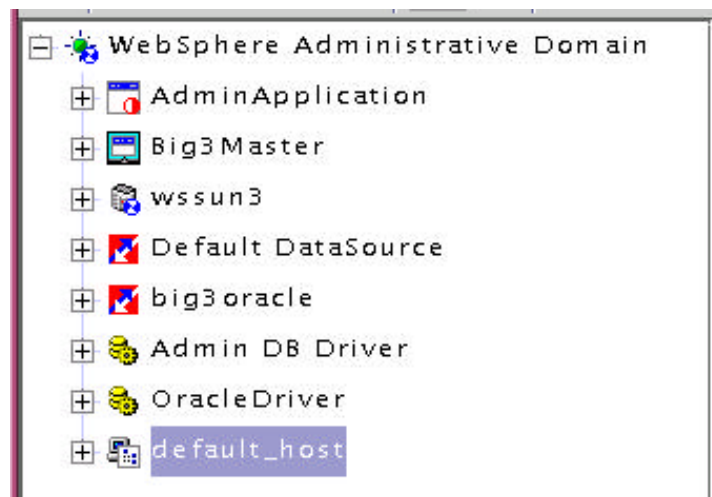
- __ G. Make note of the **Queue Name** setting, it will be referred to as `<queue_name>` for the remainder of this lab.

Queue Name:

- __ H. Click **OK** to return to the advanced property sheet.
- __ I. Click **Apply** to save the changes.
- __ J. Check the Servlet Engine **Advanced** tab settings button for the *Big3ServletEngineClone* clone and the Big3 Servlet Engine under the *Big3Server* to verify that the changes to the **Model** were automatically propagated to both of the clones.

__2. Add the alias of the Web server machine to the alias list of the virtual host.

- __ A. Locate and click the *default_host* in the **Topology** tree view.



- __ B. Under the **Advanced** tab, add entries for short name, fully qualified name, and the IP address of your NT machine to the alias list.
- __ C. Click Apply to save your changes.

__3. Start the *Big3Master* model. If it was already started, you must stop and restart it. This will start both of the cloned application servers: Big3Server and Big3clone.

__4. Monitor the `<as_root>/temp` directory until the three property files update. This could take up to 30 seconds. Use the `ls -l` command to check if the time stamp updated.

NOTE: Type **date** at a command line to display the current time on the system clock.

__5. FTP the following updated files to the `<as_root>\temp` directory of the NT Web server machine:

__ rules.properties
__ queues.properties
__ vhost.properties

NOTE: use the NT machine to **get** or **mget** the files.

__6. Update the `queues.properties` file on the Web server to reflect the remote configuration:

In the `queues.properties` file, for each clone entry that is similar to the following:

```
ose.srvgrp.<queue_name>.clone<x>.type=remote
```

add a matching host entry:

```
ose.srvgrp.<queue_name>.clone<x>.host=<SUN_Machine Name>
```

```
#IBM WebSphere Plugin Communication Queues
#Thu Jul 06 14:39:14 CDT 2000
ose.srvgrp.ibmoselink1.clonescount=2
ose.srvgrp.ibmoselink.clonescount=1
ose.srvgrp.ibmoselink.type=FASTLINK
ose.srvgrp=ibmoselink,ibmoselink1
ose.srvgrp.ibmoselink1.clone1.type=remote
ose.srvgrp.ibmoselink1.clone1.host=wssun#
ose.srvgrp.ibmoselink1.type=FASTLINK
ose.srvgrp.ibmoselink1.clone2.type=remote
ose.srvgrp.ibmoselink1.clone2.host=wssun#
ose.srvgrp.ibmoselink.clone1.type=local
ose.srvgrp.ibmoselink1.clone1.port=8994
ose.srvgrp.ibmoselink1.clone2.port=8995
ose.srvgrp.ibmoselink.clone1.port=8993
```

NOTE: Replace the `<x>` with the appropriate clone number. Be sure to do *both* clones.

__7. Save the updated `queues.properties` file.

NOTE: It is necessary to repeat **Steps 5-7** after performing any of the following activities:

- ~ Adding a new URL (Web resource) to the environment
- ~ Securing or unsecuring a URI (when you add security to a URI, a “, A” flag is added to its entry in the rules.properties file)
- ~ Configuring a new virtual host alias
- ~ Changing the queue properties of a servlet engine (name, port)
- ~ Adding a new servlet engine
- ~ Adding a new clone

Remember: You must copy the updated plugin configuration files to the temp directory of **each** Web server machine.

If we were using WebSphere Security, we would need to perform additional steps to allow the plugin to communicate with the security server.

__A. Edit the **<as_root>/properties/bootstrap.properties** file on each of the application server machines. Set the property:

```
ose.srvgrp.ibmappserve.clone1.type=remote
```

__B. Make Note of the `ose.srvgrp.ibmappserve.clone1.port` entry. It will be referred to as `<appserve_port>` for the remainder of the lab. Edit the **<as_root>/properties/bootstrap.properties** file on each of the Web server machines. Set the following properties:

```
ose.  
srvgrp.ibmappserve  
.  
clone1  
.port=<appserve_port>  
ose.srvgrp.ibmappserve.clone1.type=remote  
ose.srvgrp.ibmappserve.clone1.host=<admin server hostname>
```

NOTE: To help reduce single points of failure in a multi-node, multi-web server environment, point each Web server to a different administrative server.

__C. Stop and restart the administrative server on each of the application server machines. This will cause them to use the new **ibmappserve port** for managing security in the Web server.

__D. Stop and restart the Web servers on each of the Web server machines.

Part Two: Testing the remote OSE Configuration

- __1. Stop and restart the HTTP server on the NT machine by following the same method that you used when performing the **WAS Basic Security Using SecureWay and LDAP** lab.
- __2. Open a browser and navigate to `http://<NT_Machine_Name>/Big3/index.html`. The HTTP server on the NT machine will pass the request to the WebSphere instance on the SUN machine, and the Big3 application should load and successfully process a claim.

What you did in this exercise

You should now be able to:

- Configure a WebSphere servlet engine to accept Remote OSE requests.
- Enable remote OSE link on the HTTP Server. Configure an HTTP server to send Remote OSE requests.