Alan Altmark

Security Zones on IBM System z: Defining and Enforcing Multiple Security Zones

Introduction

Many customers wanting to do Linux® server consolidation and deployment to IBM® System z® experience challenges in balancing the security of a physical “air gap” with the low cost of consolidation on a single box: It goes against their instincts and training. Being able to understand the security implications, capabilities, advantages, and risks of virtual server consolidation helps to lower or remove barriers to growth and the decision to say “Yes” to System z.

This IBM Redpaper™ publication describes the architecture of traditional network security zones (Internet, DMZ, application, data, and maintenance) that can be securely manifested on System z through the application of best practices for System z hardware management and z/VM® virtual network configuration. This paper also discusses the Law of Unintended Consequences as it applies to shared Open Systems Adapters and HiperSockets™. We also discuss the location of firewalls, enforcement options, and an introduction to labeled security in a z/VM environment.

You learn how to properly build virtual security zones and integrate virtual servers into your existing security zones. We also discuss using the Resource Access Control Facility (RACF®) Security Server on z/VM to prevent a “red zone” server from connecting to a “green zone” network or “green zone” data.

Preparing the hardware

Before you configure z/VM, take a moment and consider the security of the entire server. If the server is not secure, then any attempt to secure z/VM might not work.

To prepare the hardware, you need to get back to basics. Ultimate power in the hands of a z/VM or z/OS® system programmer or administrator is *not* a given. People who do not have the authority to make hardware changes should not be given the privileges to do so. Authorization to issue Dynamic I/O commands in z/VM constitutes a hardware change capability (assuming the LPAR has the privilege). Be careful!

z/VM Security begins with System z security:

- Accountability is important. Obtain your own user ID on the HMC; do not share it with someone else. If you share an ID and something goes wrong, how can you prove that you did not do it?
- Limit the span of control as appropriate.
- Protect the I/O configuration by creating a separate LPAR that is authorized to modify the I/O configuration.
- Give partitions access only to devices to which they require access.

Note in Figure 1 that the dynamic I/O partition running the Hardware Configuration Definition (HCD) is not running an application workload. What if the production LPARs were compromised? Would you want either of them to have control of the I/O configuration?

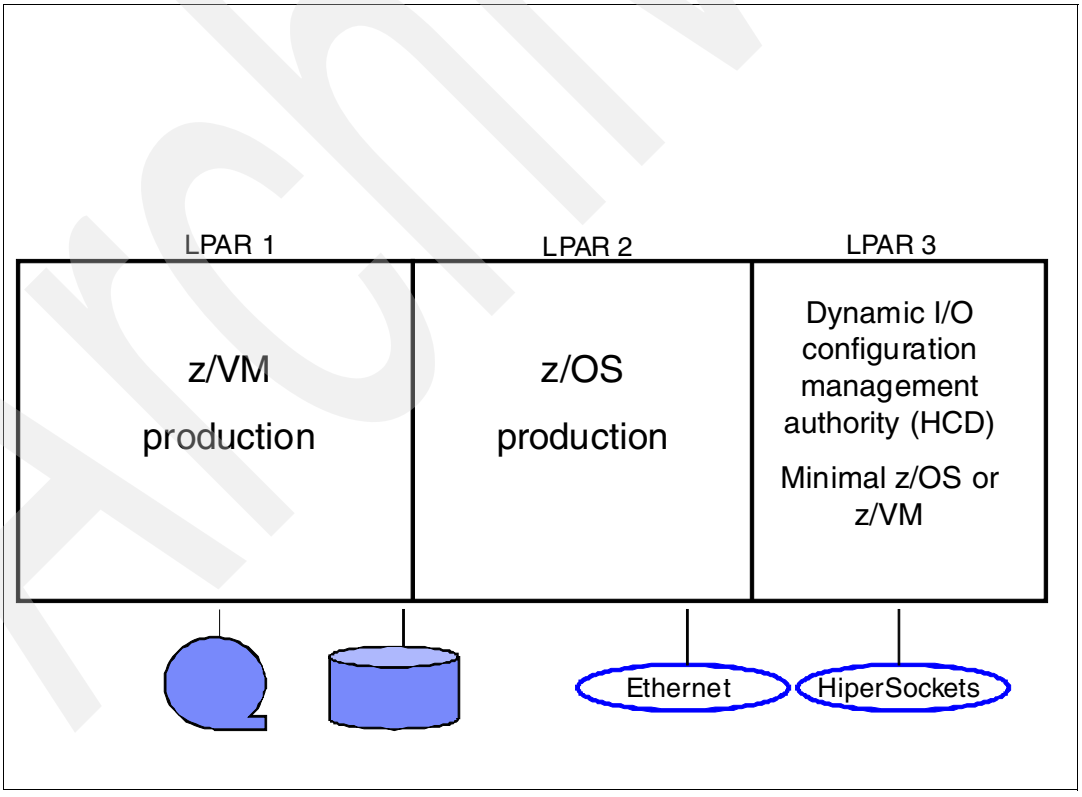

Figure 1 System z hardware security

Sharing an open system adapter (OSA) and creating a HiperSocket both create a LAN segment. Be careful about such things if it is necessary to transit a zone. You *must* have some sort of firewall technology.

Figure 2 shows how a shared OSA can create a “short circuit” between LPARs.

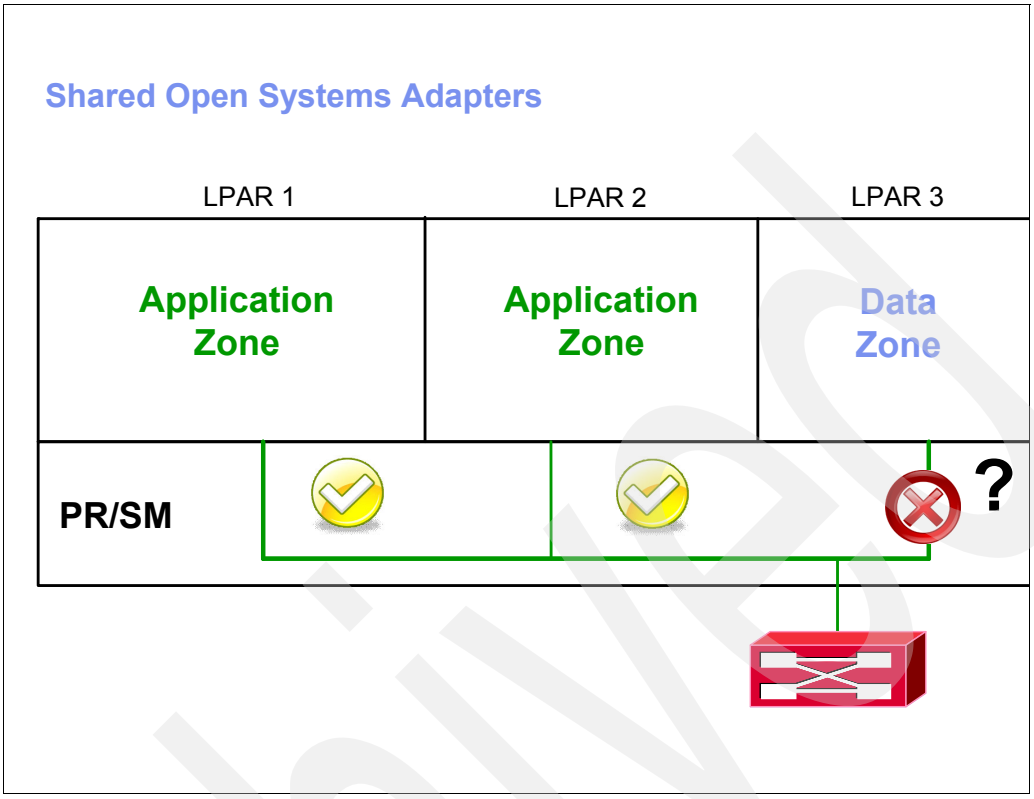

Figure 2 A shared OSA creates a “short circuit” between LPARs

The only difference between a HiperSocket and a shared OSA is the lack of a built-in bridge to the outside (see Figure 3).

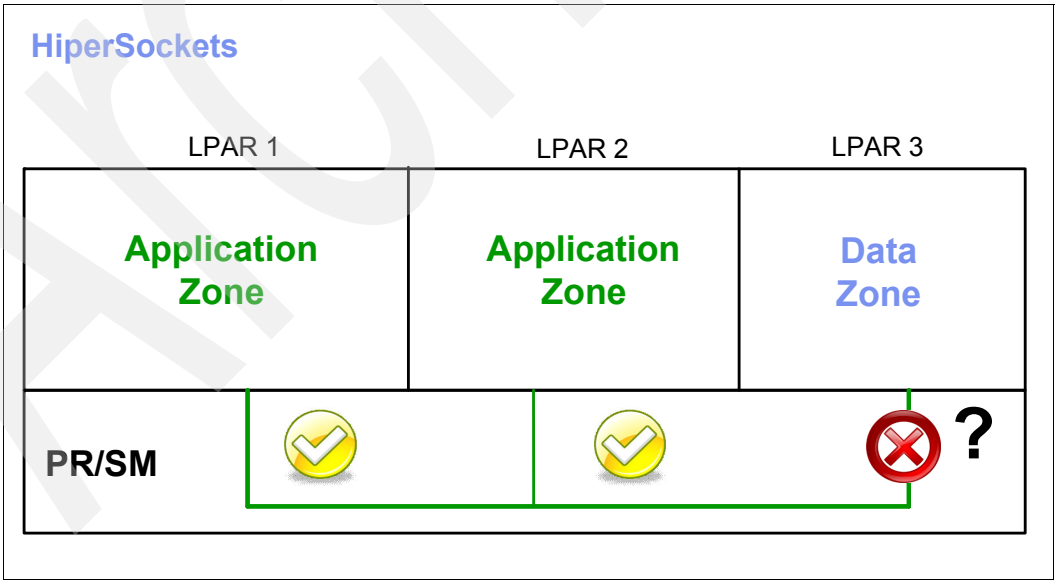

Figure 3 A HiperSocket is a LAN segment and should be treated like one

A multizone network

Figure 4 is an example of a traditional 3-tier architecture. There are four security zones. The definition of a zone is that it is an IP subnet separated from an adjacent subnets by a firewall. If you remove a firewall, you remove the “higher” of the two zones. For example, if you remove the third firewall (in Figure 4), you lose the right to host servers in the “data” zone.

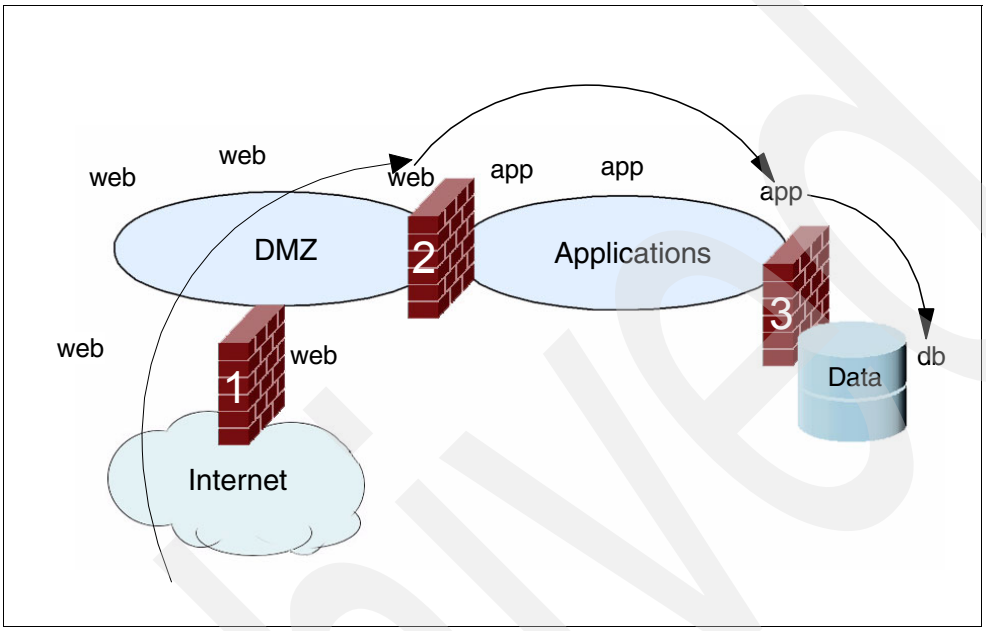

Figure 4 Traditional 3-tier application architecture

This situation might not make sense, but it is true nonetheless. It is a best practice for network security, and is required by the Payment Card Industry (PCI) security standards.

Figure 5 is an example of a network that is based on System z with external and internal firewalls.

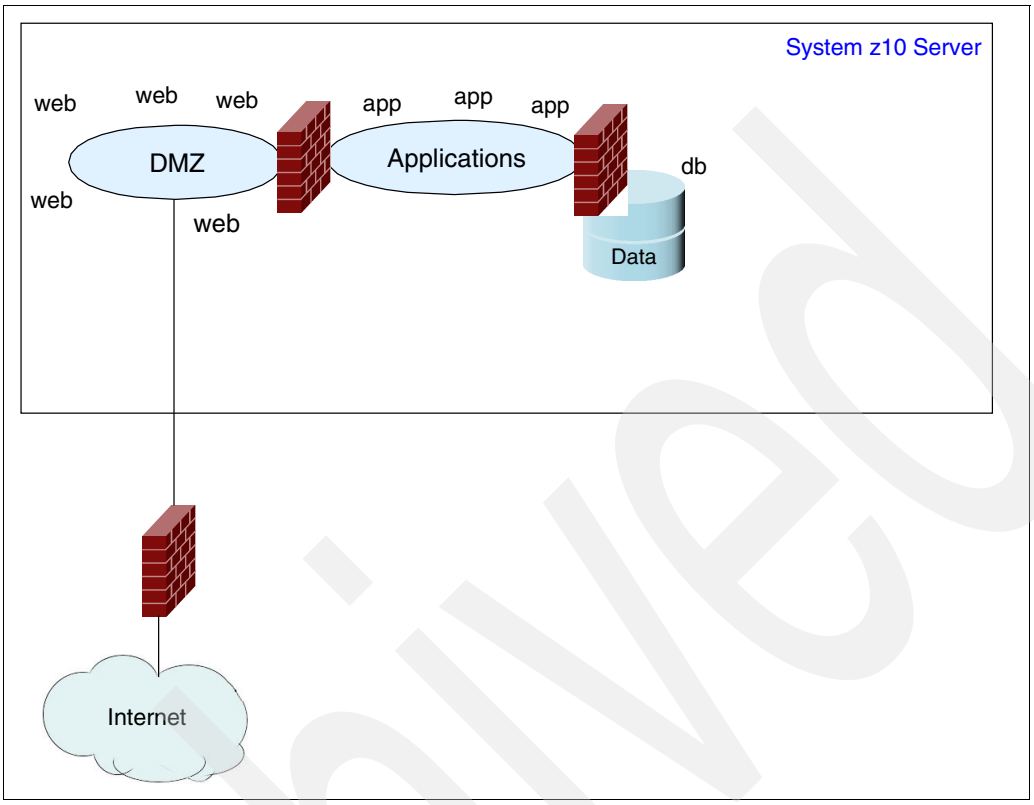

Figure 5 System z based multizone network

Firewalls

Where do you place firewalls? Figure 6 shows an example, but it is not a given that the firewalls will be running in virtual machines. Remember that network security is not the responsibility of a z/VM systems programmer; it is the responsibility of your Network Security teams. They decide what firewalls are acceptable. That decision is typically based on how they manage firewalls, as they will typically use firewall management software that can push rules to all firewalls quickly and easily. Sometimes the firewall has built-in capabilities. Unless you are part of the Network Security team, you will not be aware of all the issues.

Important: Do not surprise your Network Security team with a firewall technology of your choosing.

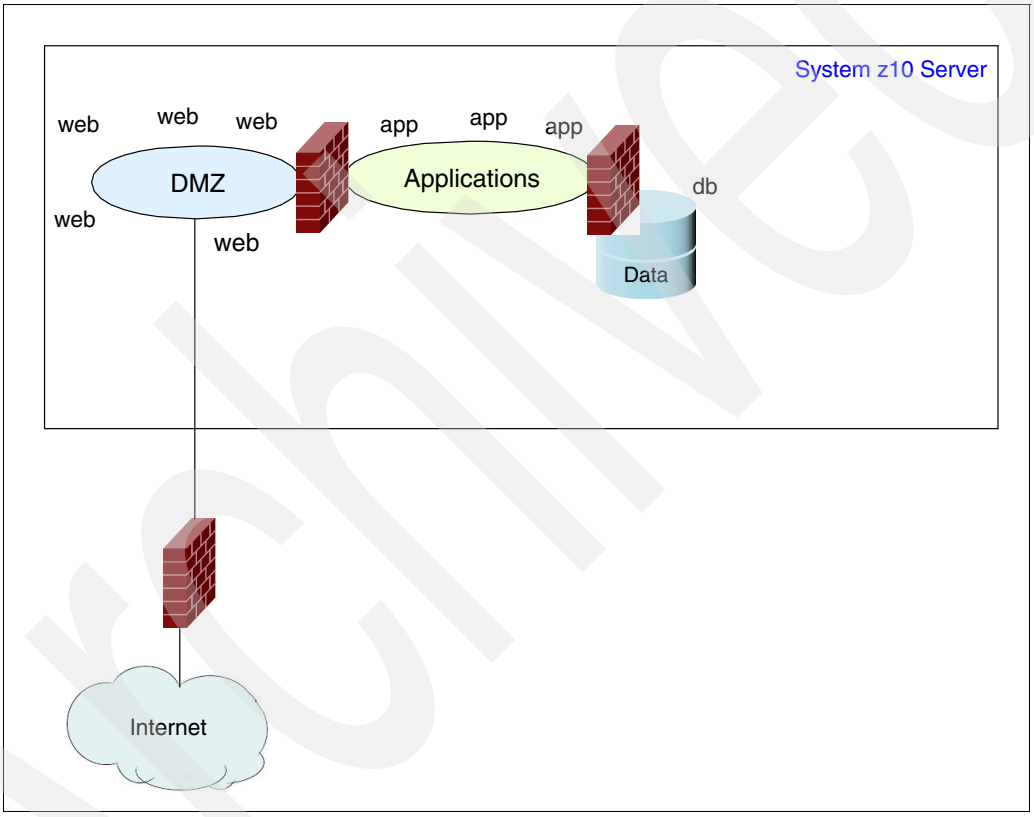

Figure 6 Onboard (internal) firewalls

Outboard, or external, firewalls are the easiest to build (see Figure 7), because you do not need to introduce unknown firewall technologies. It is not a problem if your Network Security team wants outboard firewalls. Running them outboard will increase latency, but if the transaction speed is acceptable, then the latency is acceptable. Of course, you actually have to have a measurable standard, such as one from a Service Level Agreement (SLA), order to determine if the transaction speed is acceptable.

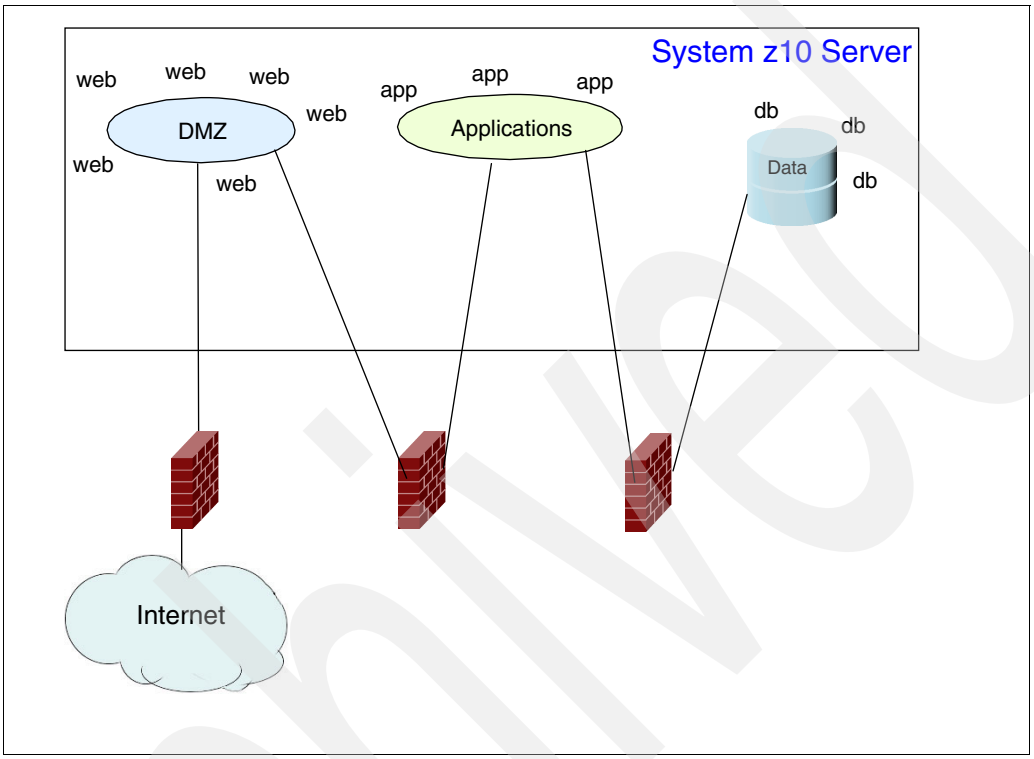

Figure 7 Outboard (external) firewalls

Combination firewalls (Figure 8) are a compromise and are all about risk management, that is, risk versus cost. If you and your security team can show value to onboard firewalls, then go ahead and use them. But remember that you have the combination firewall option. The workloads you really should be concerned about are associated with the web servers, application servers, and the database.

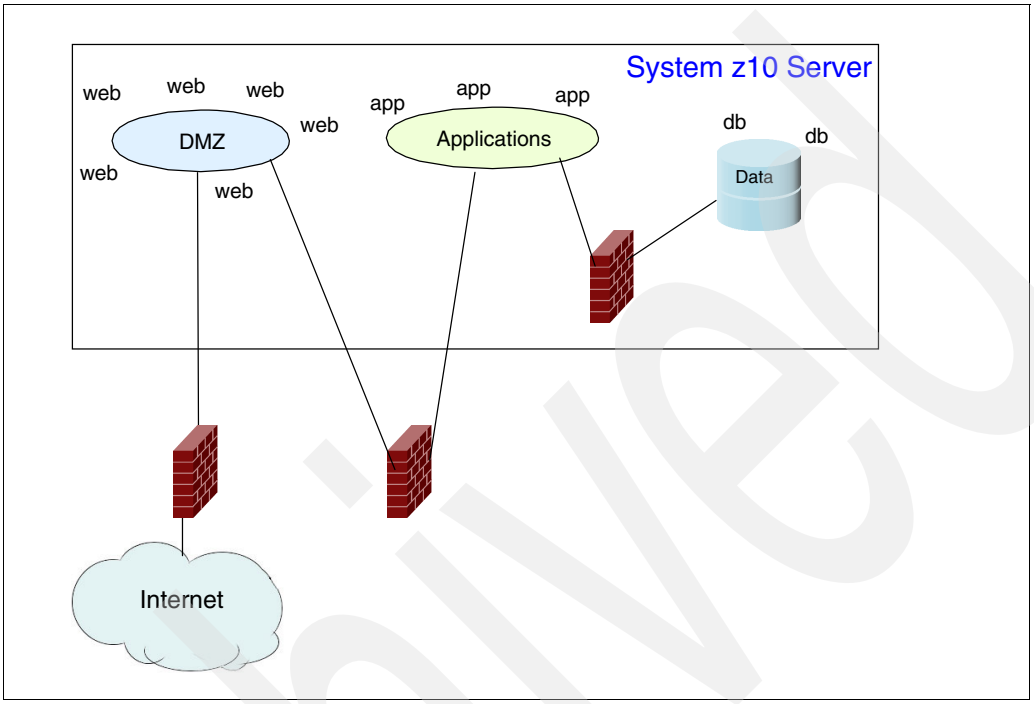

Figure 8 Combination firewalls

One implementation of a combination firewalls solution would be guest LANs with HiperSockets, as shown in Figure 9.

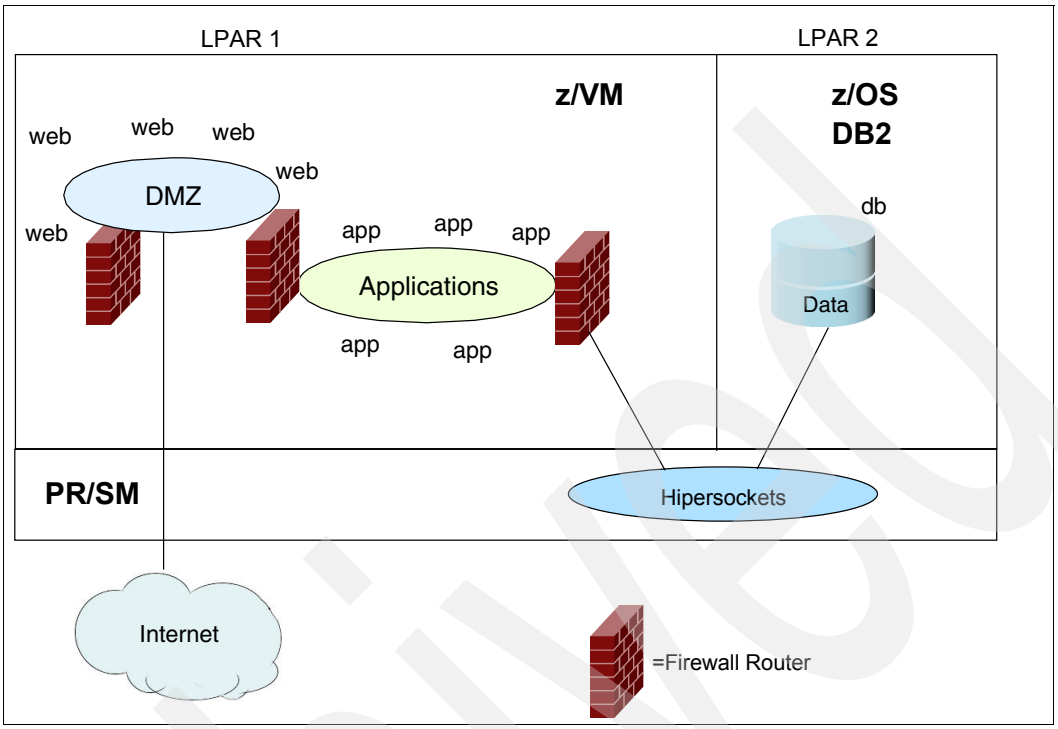

Figure 9 Guest LANs with HiperSockets

Another implementation of a combination firewalls solution would be HiperSockets and IBM z/OS packet filters, as shown in Figure 10

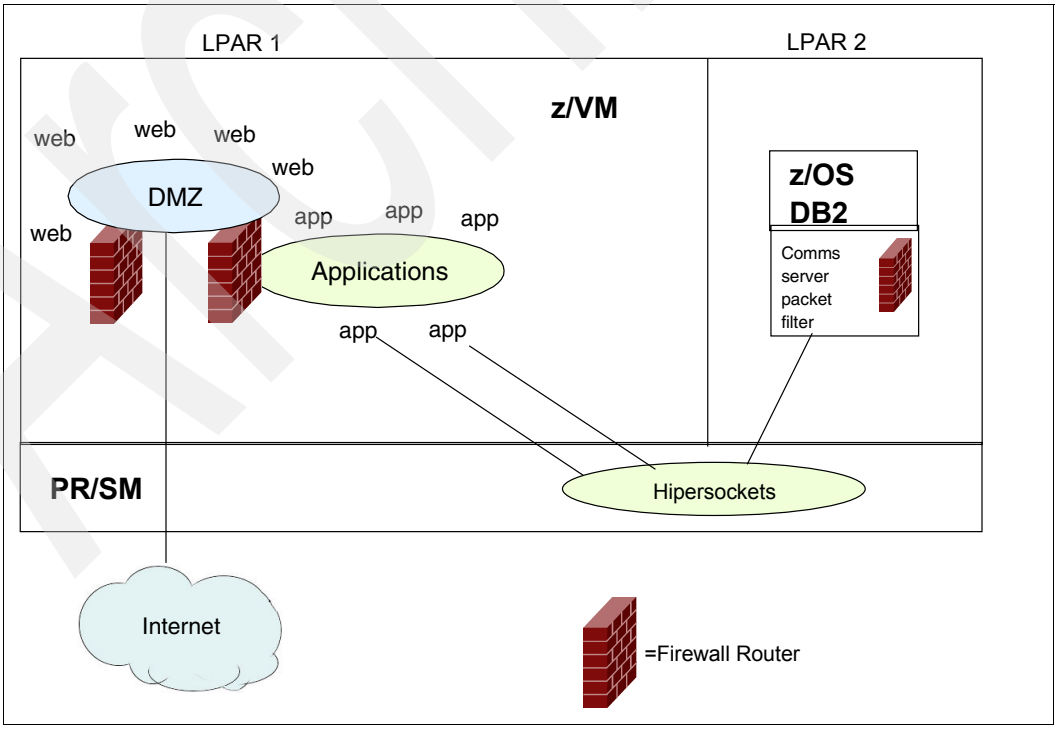

Figure 10 HiperSockets and z/OS packet filters

We are using the built-in packet filtering technology of z/OS and dedicated HiperSockets (which allows us to use QDIO Assist). Of course, all usage of z/OS packet filters must be negotiated.

Another implementation of a combination firewall solution is shown in Figure 11.

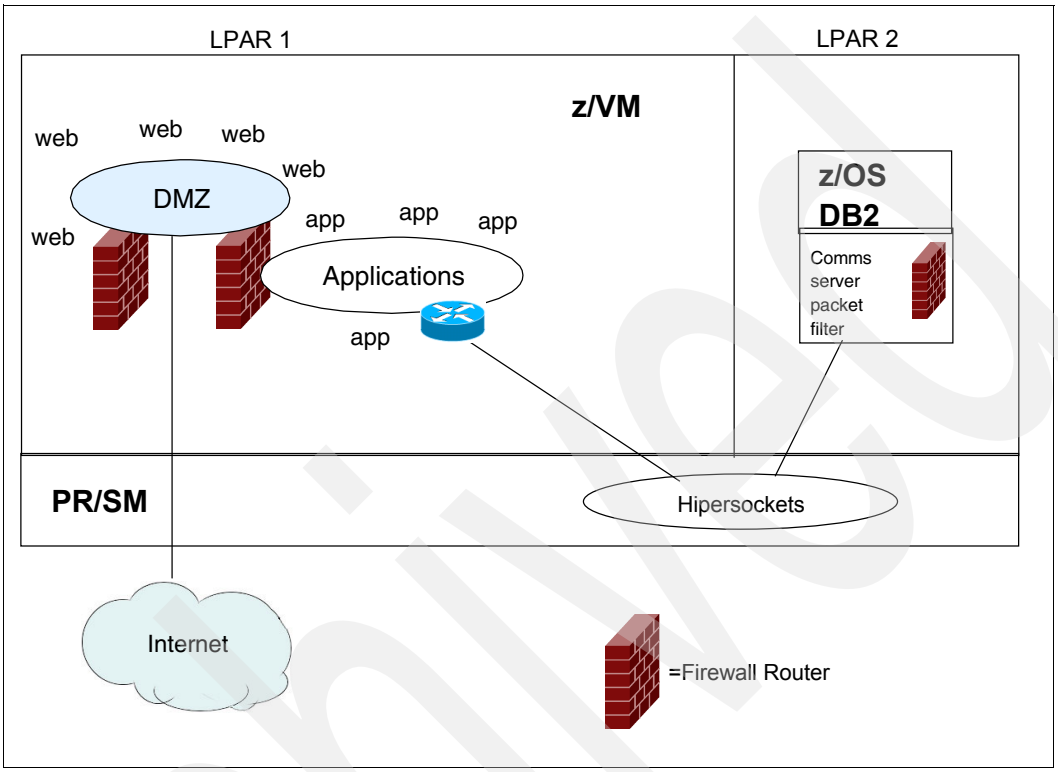

Figure 11 HiperSockets and z/OS packet filters

This implementation is the same as shown in Figure 10 on page 9, but all applications to z/OS traffic is funneled through a virtual router with no firewall. This is a bad idea and does not add any value.

Virtual switches, VLANs, and traffic separation

A switch creates LANs and routes traffic. It turns ports on and off, assigns a port to a LAN segment, and provides LAN “sniffer” ports. A switch is similar to a hub, but with better technology, more function, and more expensive.

With a hub, the ports are physically connected. Each port sees all the other ports. All ports are at the same speed. A hub can be considered “dumb.”

Conversely, with a switch, the ports are logically connected based on administrative settings in the switch, so a switch is considered “smart”.

Figure 12 shows two types of switches.

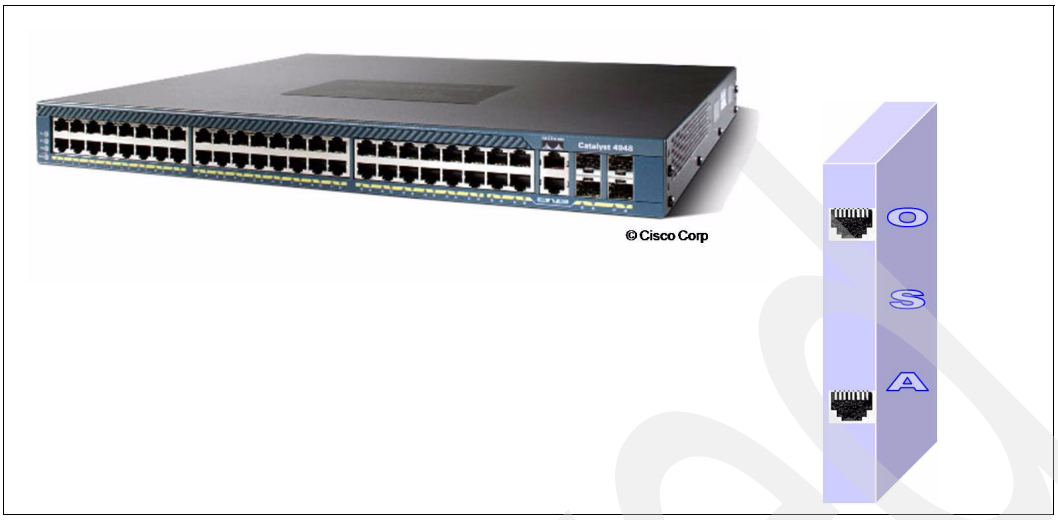

Figure 12 Sample switches

What do you do when all the ports on a switch are filled? You connect the switch to another switch. If you run out of ports, you daisy chain (trunk) the switch to another switch using a *trunk port.*

A trunk port carries Ethernet frames with an extra piece of information called the *VLAN ID tag*. This tag tells the target switch which VLAN (LAN segment) to which the frame belongs. The original switch is responsible for adding the tag. The tag is removed before it is sent out on an *access port*. Tags are not accepted from an access port; they will be treated as a malformed frame. Figure 13 demonstrates a trunk port (indicated by a "T") versus an access port.

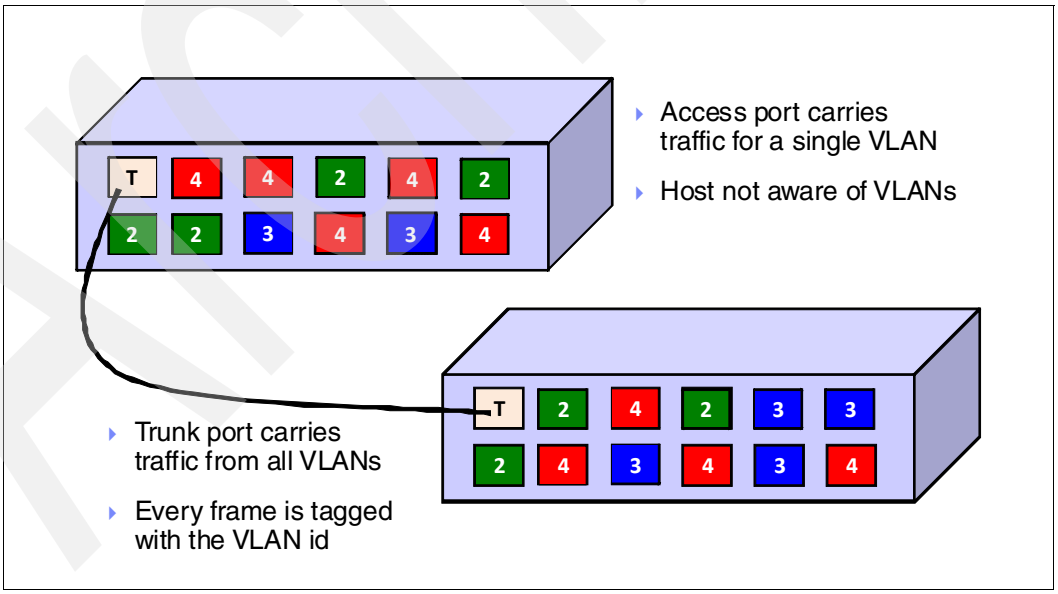

Figure 13 Trunk port versus an access port

All we are doing with a VSWITCH is putting it in place of a real switch. It has all the rights, privileges, and responsibilities of a real switch. Figure 14 shows that the trunk port on the physical switch can be restricted to a subset of all possible VLANs carried on the switch, that is, there may be some other VLANs that you cannot see.

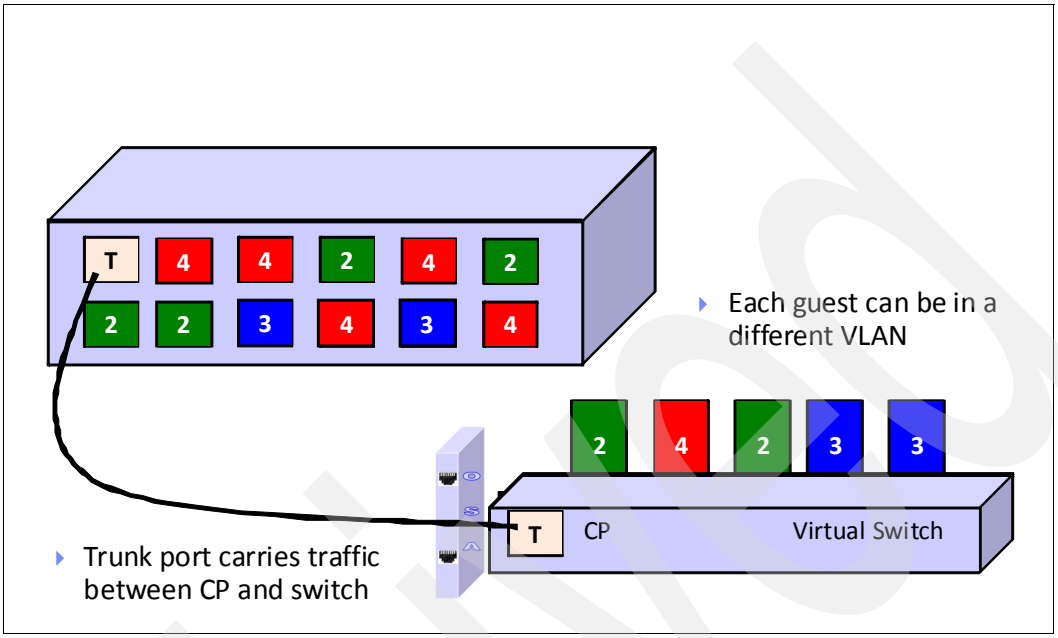

Figure 14 Physical Switch to Virtual Switch

A VLAN unaware VSWITCH plugs into an access port. It sees only a single LAN segment. Think of it as monochromatic or color blind. It does not know about other LAN segments (Figure 15).

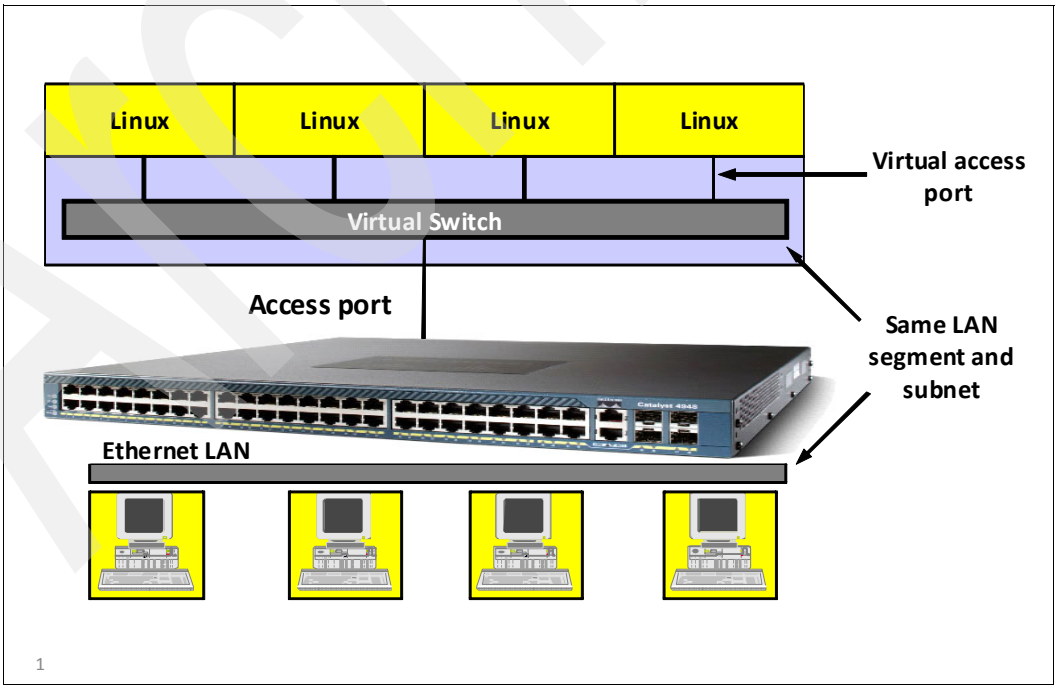

Figure 15 z/VM Virtual Switch: VLAN unaware

In Figure 16, the VSWITCH is plugged into a trunk port and can receive data from or send data to any VLAN that has been authorized on that port by the switch administrator. In fact, the VSWITCH is required to tag outgoing frames and remove tags from incoming frames.

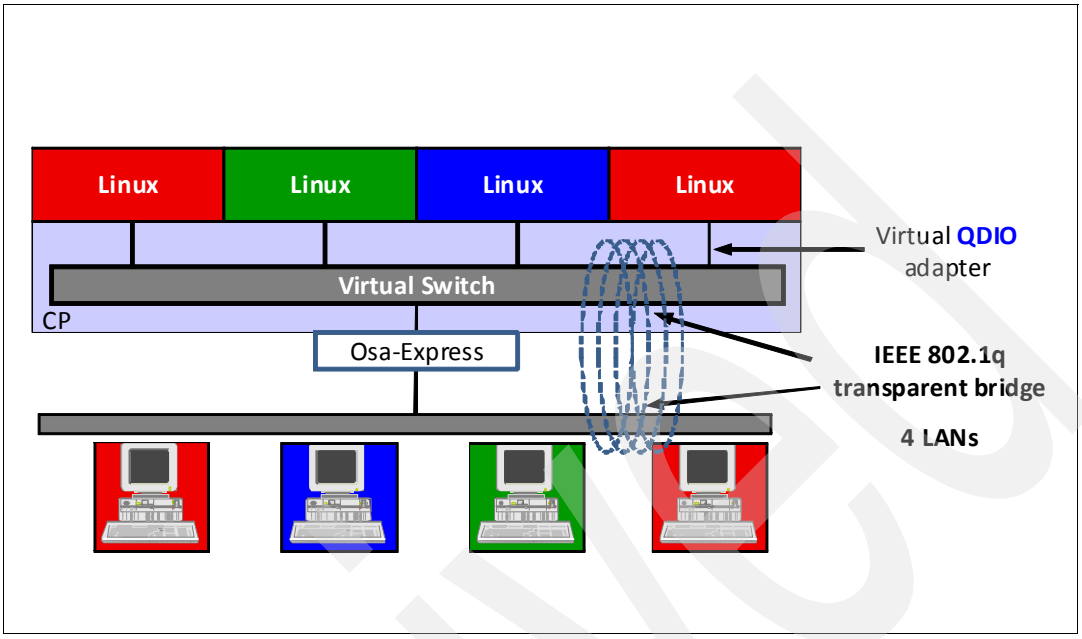

Figure 16 z/VM Virtual Switch VLAN assignment

Using a VSWITCH without an OSA acts a better Guest LAN than an actual Guest LAN (DEFINE LAN) because it has more controls, as shown in Figure 17. In particular, you can enforce VLAN usage restrictions on virtual trunk ports.

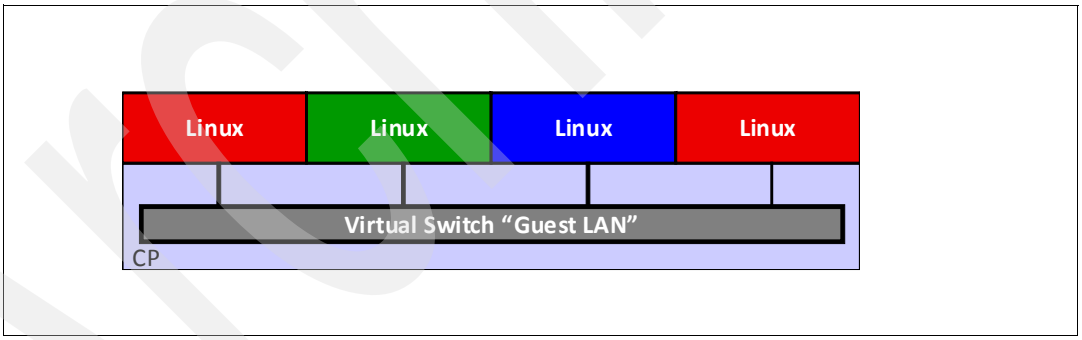

Figure 17 z/VM Virtual Switch: VLAN aware and no OSA

One VSWITCH carrying data for two LAN segments is shown in Figure 18.

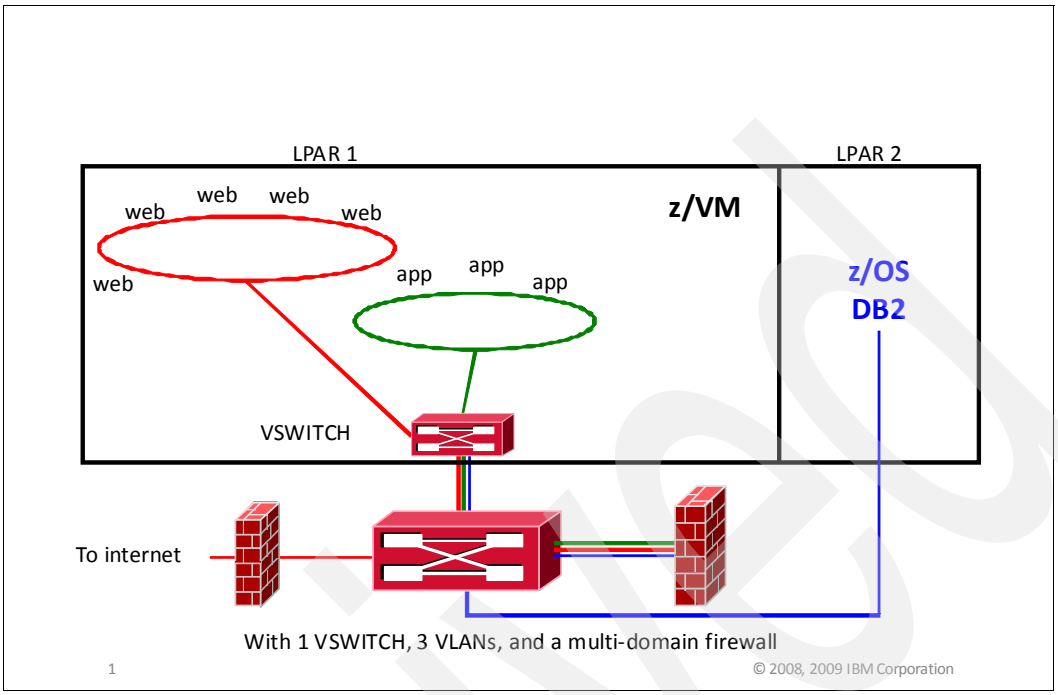

Figure 18 Network with VSWITCH (with VLANs)

Figure 19 shows two VSWITCHes. There is physical traffic separation. Of course, for high availability, you need two OSAs per VSWITCH. In theory, they could back up each other, but then you would be unable to use link aggregation.

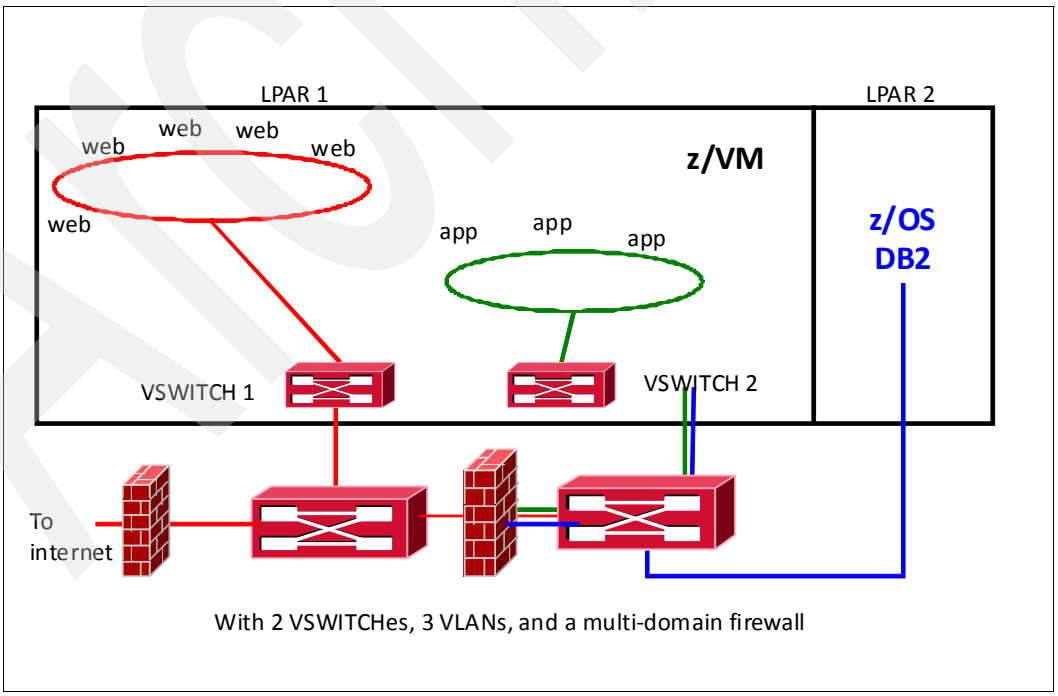

Figure 19 Multi-zone network with VSWITCH (no VLANs)

Enforcing the rules with Resource Access Control Facility

Resource Access Control Facility (RACF) creates objects in classes. The class is the name space. Each instance of an object in a class represents another VM resource (disk, VSWITCH, user, and so on).

Protecting a VSWITCH using RACF includes having access control through the VMLAN class in RACF. The VMLAN class allows a guest system to establish a connection to a VSWITCH. The VMLAN class:

- Uses SYSTEM.name or SYSTEM.name.vlanid
- Uses owner.name for Guest LANs

The command used to permit access to a VSWITCH is:

```
PERMIT SYSTEM.VSW01 CLASS(VMLAN) ID(ALAN) ACCESS(UPDATE)
```

z/VM virtualizes a LAN sniffer to capture network traffic on a z/VM Guest LAN or VSWITCH. This action helps capture network data to resolve virtual networking problems. Using this sniffer function requires CONTROL access.

Port isolation security provides the ability to restrict guest-to-guest communications within a VSWITCH by exploiting OSA-Express QDIO data connection isolation. To do this task, run the following command:

```
SET  VSWITCH name ISOLATE
```

After this command is done, guests cannot talk to each other. On IBM System z10® and later, VSWITCH port isolation also activates QDIO data connection isolation, disabling the “short circuit” in the OSA. There is no cross-talk on shared OSA to and from the VSWITCH.

Another way to enforce the rules using the VMLAN class is to turn off backchannel communications. Backchannel communications allow users with high security clearance within a specific Linux server to bypass the security rules and polices that apply to the whole network infrastructure.

To turn off backchannel communications:

- Do not allow the creation of user-defined Guest LANs. To prohibit users from defining any Guest LANs on a given system, set the limit to zero (0). For example, the SET VMLAN LIMIT TRANSIENT 0 command would prevent general users from creating Guest LANs.
- Do not allow class G users the ability to define virtual channel-to-channel (CTC) interfaces from within their servers. After you have already given your Linux virtual servers class G privileges, remove the DEFINE command from class G users by running the following command:

  ```
  MODIFY COMMAND DEFINE IBMCLASS G PRIVCLASS M
  ```

- Always use explicit Inter-User Communications Vehicle (IUCV) authorization in the directory. Do not use the IUCV ALLOW or IUCV ANY commands.
- Remove the ability from class G users to set secondary consoles by running the following command:

  ```
  MODIFY COMMAND SET SECUSER IBMCLASS G PRIVCLASS M
  ```

Additionally, you could also use the following utilities to turn off backchannel communications:

- You can use the Virtual Machine Communication Framework (VMCF) by running the following command:

  ```
  MODIFY DIAGNOSE DIAG068 IBMCLASS G PRIVCLASS M
  ```

- ESA/XC mode address space sharing (ADRSPACE PERMIT)
- Discontiguous Shared Segments (DCSS)

While these methods are all efficient and considered essential to maintain system's security, they are all discretionary rules that need to be carefully maintained to avoid security breaches. And then there are new interfaces that can be added in an authorized program analysis report (APAR). You have to wonder how you can keep up with it all.

To address this issue, the implementation of RACF support for Mandatory Access Controls (MAC) would be a step in the right direction. MAC is a security policy that governs which subject, or users, can access which resources and in what way. MAC can restrict access to an object in regards to three things:

- The security label of the subject.
- The security label of the object.
- The type of access the subject requires for the task being performed.

If the MAC criteria are met, z/VM proceeds with the discretionary access control where appropriate. With MAC implemented, it is possible to separate all the virtual resources, or objects, in their security zones, and give system administrators more flexibility managing their devices. MAC guarantees that only subjects previously authorized to use a resource within a security zone would have the privilege to do so.

On Linux, AppArmor by Novell and Security-Enhanced Linux (SELinux) on Red Hat Enterprise Linux provide the same functionality for Linux as MAC.

MAC guarantees that only subjects previously authorized to use a resource within a security zone would have the privilege to do so, as shown in Figure 20.

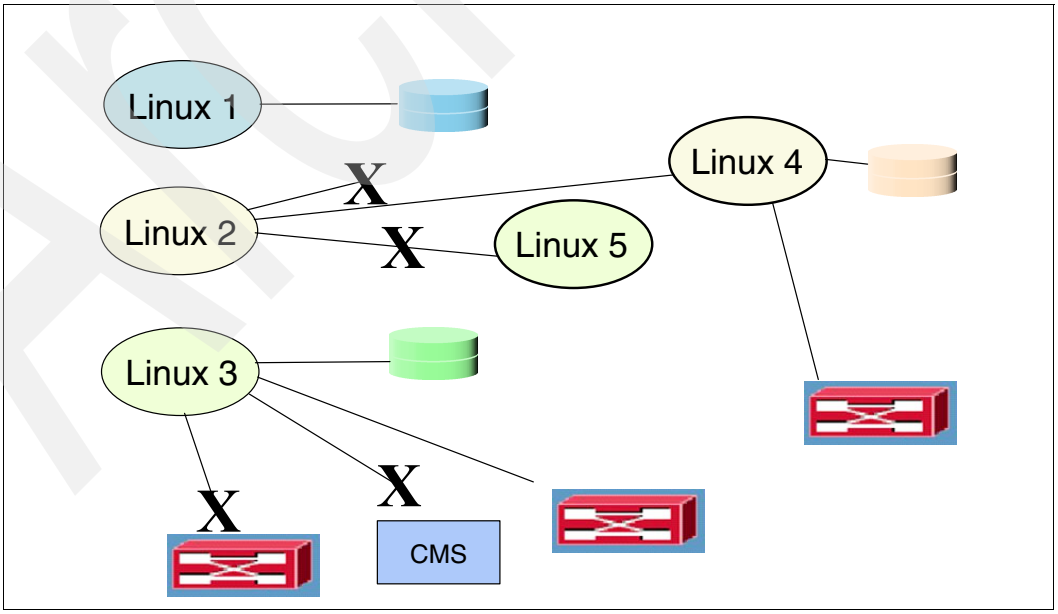

Figure 20 Multi-zone z/VM LPAR with RACF enforcement

Because the security administrator overrides the user and the system programmer, Mandatory Access Controls override user controls and user or administrator given permissions. Users are assigned to one or more named projects. Minidisks, guest LANs, VSWITCHes, VLAN IDs, NSSes, DCSSes, and spool files all represent data in those same projects. Users can only access data in their assigned projects.

When using multizoning with RACF, a security label combines the concepts of security clearance (secret, top secret, and eyes only) and data partitioning (which applies to disks, networks, and other users).

“Labeled security” is an old, well-understood concept in the industry and is part of the z/VM Common Criteria certification. As of the writing of this paper, no other security product for z/VM provides this capability.

While there is no real use for security clearance on virtual servers, it could be used to prevent read/write (RW) access to data by people who are authorized only to look at data; it does not prevent copying of the data, though.

The commands used to create security levels and data partitions are shown in Example 1.

Example 1 Create security levels and data partitions

```
1 RDEFINE SECDATA SECLEVEL ADDMEM(DEFAULT/100)
2 RDEFINE SECDATA CATEGORY ADDMEM(INTERNET DMZ APPS DATA COMMON)
3 RDEFINE SECLABEL PUBLIC SECLEVEL(DEFAULT)ADDCATEGORY(COMMON) UACC(NONE)
4 RDEFINE SECLABEL RED SECLEVEL(DEFAULT)ADDCATEGORY(DMZ COMMON) UACC(NONE)
5 RDEFINE SECLABEL GREEN SECLEVEL(DEFAULT) ADDCATEGORY(APPS COMMON) UACC(NONE)
6 RDEFINE SECLABEL BLUE SECLEVEL(DEFAULT) ADDCATEGORY(DATA COMMON) UACC(NONE)
```

In Example 1, line 1 shows the command used to create a security level and name it `DEFAULT` with `100` as the level of security clearance. The number of security levels and the hierarchy you use depend on how your infrastructure is set up and what your security requirements are. In line 2, four categories are created, `INTRANET`, `DMZ APPS`, `DATA`, and `COMMON`, which are used to represent the resources accessible by the external network, the DMZ, applications, and other areas. Lines 3 through 6 group security levels and categories together. After the security levels and data partitions are created, you can assign virtual machines their security labels by using SECLABEL. It may seem unnecessary to use PERMIT and ALTUSER, but you need to assign the user a default label (otherwise, the user has no label, in which case SETROPTS controls what happens).

Example 2 shows the commands that are used to assign the security labels to virtual machines.

Example 2 Assigning security labels

```
PERMIT RED CLASS(SECLABEL) ID(LXHTTP01) ACCESS(READ)
ALTUSER LXHTTP01 SECLABEL(RED)
PERMIT GREEN CLASS(SECLABEL) ID(LXWAS001) ACCESS(READ)
ALTUSER LXWAS001 SECLABEL(GREEN)
```

Sometimes a server provides services to all users, which exempts a server from label checking. You can set this configuration by assigning the label SYSNONE to the system servers (see Example 3). SYSNONE is predefined and should be used only for resources that have no classified data content. Do not confuse SYSNONE with NONE, SYSHIGH, or SYSLOW, which produce certain results for a label comparison; SYSNONE causes the label check to be bypassed. (Label checks are not character string checks, but an analysis of the security level and categories that comprise the label.)

Example 3 Using SYSNONE

```
PERMIT SYSNONE CLASS(SECLABEL) ID(TCPIP) ACCESS(READ)
ALTUSER TCPIP SECLABEL(SYSNONE)
```

So, by design, you would either protect or not protect your resources. There is little middle ground. If you intend to activate the TERMINAL or VMSEGMT classes, those resources all need SECLABELs.

To assign labels to resources, for example, your minidisk, Guest LANs, and virtual switches, your commands should look similar to the ones shown in Example 4.

Example 4 Assigning labels to resources

```
RALTER VMMDISK LXHTTP01.201 SECLABEL(RED)
RALTER VMLAN SYSTEM.NET1 SECLABEL(RED)
RALTER VMLAN SYSTEM.NET2.0307 SECLABEL(GREEN)
```

To activate RACF protection, you can either run the commands in Example 5 or Example 6. In Example 5, if the resource does not have a SECLABEL, a warning is issued and SECLABELs are ignored.

Example 5 Activating the RACF protection so a warning is issued if the resource does not have a SECLABEL

```
SETROPTS CLASSACT(SECLABEL VMMDISK VMLAN)
SETROPTS RACLIST(SECLABEL)
SETROPTS MLACTIVE(WARNINGS)
```

Example 6 is more secure because if the resource does not have a SECLABEL, the command fails.

Example 6 Activating the RACF protection so the command fails with no SECLABEL

```
SETROPTS MLACTIVE(FAILURES)
```

Summary

In summary, when creating security zones in z/VM, keep in mind the following points:

- Have a network architect check your network design.
- Do not complain about firewalls.
- Optimize host-resident firewalls later.
- Protect your hardware, data, servers, and your company.

Reference information

For more information about the topics discussed in this paper, refer the following online resources:

- Presentations on this and other z/VM related security subjects:

  http://www.VM.ibm.com/devpages/altmarka/present.html
- System z security:

  http://www.ibm.com/systems/z/advantages/security/
- z/VM home page:

  http://www.vm.ibm.com
- *z/VM Secure Configuration Guide*, found at:

  http://publibz.boulder.ibm.com/epubs/pdf/hcss0b30.pdf
- z/VM security resources:

  http://www.VM.ibm.com/security

The author who wrote this paper

This paper was produced by a technical specialist working at the International Technical Support Organization, Poughkeepsie Center.

Alan Altmark is a z/VM and Linux on System z IT Consultant who is based at IBM Poughkeepsie.

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks
- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks® publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

This document REDP-4652-00 was created or updated on February 16, 2011.

Send us your comments in one of the following ways:
- Use the online **Contact us** review Redbooks form found at:
  ibm.com/redbooks
- Send your comments in an email to:
  redbooks@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400 U.S.A.

Trademarks