

Table of Contents

<u>Securing Networks with Private VLANs and VLAN Access Control Lists</u>	1
<u>Document ID: 10601</u>	1
<u>Introduction</u>	1
<u>Before You Begin</u>	1
<u>Conventions</u>	1
<u>Prerequisites</u>	1
<u>Components Used</u>	2
<u>Background Information</u>	2
<u>Importance of Enforcing a Proper Trust Model</u>	2
<u>Private VLANs</u>	3
<u>VLAN Access Control Lists</u>	4
<u>Known Limitations of VACLs and PVLANS</u>	4
<u>Example Case Studies</u>	5
<u>Pass-Through DMZ</u>	5
<u>External DMZ</u>	11
<u>VPN Concentrator in Parallel to Firewall</u>	14
<u>Related Information</u>	22

Securing Networks with Private VLANs and VLAN Access Control Lists

Document ID: 10601

Introduction

Before You Begin

- Conventions

- Prerequisites

- Components Used

- Background Information

Importance of Enforcing a Proper Trust Model

Private VLANs

VLAN Access Control Lists

Known Limitations of VACLs and PVLANS

Example Case Studies

- Pass-Through DMZ

- External DMZ

- VPN Concentrator in Parallel to Firewall

Related Information

Introduction

One of the key factors to building a successful network security design is to identify and enforce a proper trust model. The proper trust model defines who needs to talk to whom and what kind of traffic needs to be exchanged; all other traffic should be denied. Once the proper trust model has been identified, then the security designer should decide how to enforce the model. As more critical resources are globally available and new forms of network attacks evolve, the network security infrastructure tends to become more sophisticated, and more products are available. Firewalls, routers, LAN switches, intrusion detection systems, AAA servers, and VPNs are some of the technologies and products that can help enforce the model. Of course, each one of these products and technologies plays a particular role within the overall security implementation, and it is essential for the designer to understand how these elements can be deployed.

Before You Begin

Conventions

For more information on document conventions, see the Cisco Technical Tips Conventions.

Prerequisites

This document describes PVLAN configurations on switches running CatOS only. For side-by-side configuration examples of PVLANS on switches running Cisco IOS and CatOS, refer to the document [Configuring Isolated Private VLANs on Catalyst Switches](#).

Not all switches and software versions support PVLANS. Refer to [Private VLAN Catalyst Switch Support Matrix](#) to determine whether your platform and software version supports PVLANS.

Components Used

This document is not restricted to specific software and hardware versions.

Background Information

Identifying and enforcing a proper trust model seems to be a very basic task, but after several years of supporting security implementations, our experience indicates that security incidents are often related to poor security designs. Usually these poor designs are a direct consequence of not enforcing a proper trust model, sometimes because what is just necessary is not understood, other times just because the technologies involved are not fully understood or are misused.

This document explains in detail how two features available in our Catalyst switches, Private VLANs (PVLANS) and VLAN Access Control Lists (VACLs), can help ensure an adequate trust model in both enterprise as well as service provider environments.

Importance of Enforcing a Proper Trust Model

An immediate consequence of not enforcing an adequate trust model is that the overall security implementation becomes less immune to malicious activities. Demilitarized Zones (DMZs) are commonly implemented without enforcing the right policies, thus facilitating the activity of a potential intruder. This section analyzes how DMZs are often implemented and the consequences of a poor design. We will later explain how to mitigate, or in the best case avoid, these consequences.

Usually, DMZ servers are only supposed to process incoming requests from the Internet, and eventually initiate connections to some back-end servers located at an inside or other DMZ segment, such as a database server. At the same time, DMZ servers are not supposed to talk to each other or initiate any connections to the outside world. This clearly defines the necessary traffic flows in a simple trust model; however, we often see this kind of model not adequately enforced.

Designers usually tend to implement DMZs using a common segment for all servers without any control over the traffic between them. For example, all servers are located in a common VLAN. Since nothing is controlling the traffic within the same VLAN, if one of the servers is compromised, then the same server can be exploited to source an attack to any of the servers and hosts in the same segment. This clearly facilitates the activity of a potential intruder conducting a port redirection or Application Layer attack.

Typically, firewalls and packet filters are only used to control incoming connections, but nothing is usually done to restrict connections originated from the DMZ. Some time ago there was a well-known vulnerability in a cgi-bin script that allowed an intruder to begin an X-term session by just sending an HTTP stream; this is traffic that should be allowed by the firewall. If the intruder was lucky enough, he or she could use another treat to get a root prompt, typically some kind of buffer overflow attack. Most of the times these kinds of problems can be avoided by enforcing a proper trust model. First, servers are not supposed to talk to each other, and second no connections should be originated from these servers to the outside world.

The same comments apply to many other scenarios, going from any regular un-trusted segment up to server farms at application service providers.

PVLANS and VACLs on Catalyst switches can help ensure a proper trust model. PVLANS will help by restricting the traffic between hosts in a common segment, while VACLs will contribute by providing further control over any traffic flow originated or destined to a particular segment. These features are discussed in the following sections.

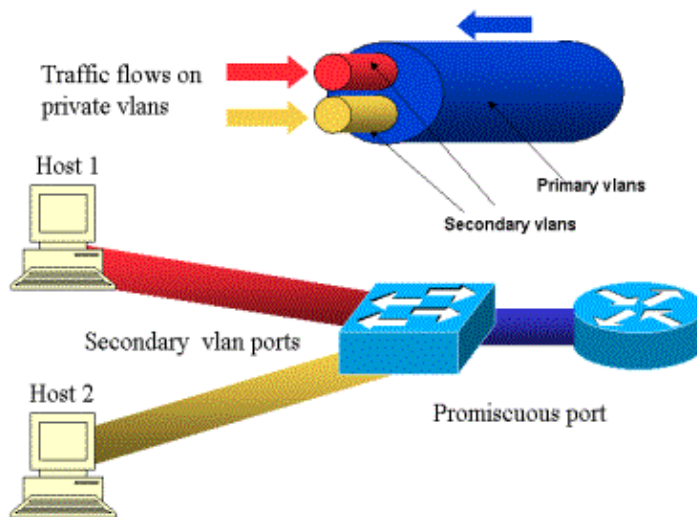
Private VLANs

PVLANs are available on the Catalyst 6000 running CatOS 5.4 or later, on the Catalyst 4000, 2980G, 2980G-A, 2948G, and 4912G running CatOS 6.2 or later.

From our perspective, PVLANs are a tool that allows segregating traffic at Layer 2 (L2) turning a broadcast segment into a non-broadcast multi-access-like segment. Traffic that comes to a switch from a promiscuous port (that is, a port that is capable of forwarding both primary and secondary VLANs) is able to go out on all the ports that belong to the same primary VLAN. Traffic that comes to a switch from a port mapped to a secondary VLAN (it can be either an isolated, a community, or a two-way community VLAN) can be forwarded to a promiscuous port or a port belonging to the same community VLAN. Multiple ports mapped to the same isolated VLAN cannot exchange any traffic.

The following image shows the concept.

Figure 1: Private VLANs



The primary VLAN is represented in blue; the secondary VLANs are represented in red and yellow. Host-1 is connected to a port of the switch that belongs to the secondary VLAN red. Host-2 is connected to a port of the switch that belongs to the secondary VLAN yellow.

When a host is transmitting, the traffic is carried in the secondary VLAN. For example, when Host-2 transmits, its traffic goes on VLAN yellow. When those hosts are receiving, the traffic comes from the VLAN blue, which is the primary VLAN.

The ports where routers and firewalls are connected are promiscuous ports because those ports can forward traffic coming from every secondary VLAN defined in the mapping as well as the primary VLAN. The ports connected to each hosts can only forward the traffic coming from the primary VLAN and the secondary VLAN configured on that port.

The drawing represents the private VLANs as different pipes that connect routers and hosts: the pipe that bundles all the others is the primary VLAN (blue), and the traffic on VLAN blue flows from the routers to the hosts. The pipes internal to the primary VLAN are the secondary VLANs, and the traffic traveling on those pipes is from the hosts towards the router.

As the image is showing, a primary VLAN can bundle one or more secondary VLANs.

Earlier in this document we said PVLANS help enforce the proper trust model by simply ensuring the segregation of hosts within a common segment. Now that we know more about Private VLANs, let us see how this can be implemented in our initial DMZ scenario. Servers are not supposed to talk to each other, but they still need to talk to the firewall or router to which they are connected. In this case, servers should be connected to isolated ports while routers and firewalls should be attached to promiscuous ports. By doing this, if one of the servers is compromised, the intruder won't be able to use the same server to source an attack to another server within the same segment. The switch will drop any packet at wire speed, without any performance penalty.

Another important note is that this kind of control can only be implemented at the L2 device because all servers belong to the same subnet. There is nothing a firewall or router can do since servers will try to communicate directly. Another option is to dedicate a firewall port per server, but this is likely too expensive, difficult to implement, and does not scale.

In a later section, we describe in detail some other typical scenarios in which you can use this feature.

VLAN Access Control Lists

VACLs are available on the Catalyst 6000 series running CatOS 5.3 or later.

VACLs can be configured on a Catalyst 6500 at L2 without the need for a router (you only need a Policy Feature Card (PFC)). They are enforced at wire speed so there is no performance penalty in configuring VACLs on a Catalyst 6500. Since the lookup of VACLs is performed in hardware, regardless of the size of the access list, the forwarding rate remains unchanged.

VACLs can be mapped separately to primary or secondary VLANs. Having a VACL configured on a secondary VLAN allows filtering the traffic originated by hosts without touching the traffic generated by routers or firewalls.

By combining VACLs and Private VLANs it is possible to filter traffic based on the direction of the traffic itself. For example, if two routers are connected to the same segment as some hosts (servers for example), VACLs can be configured on secondary VLANs so that only the traffic generated by the hosts is filtered while the traffic exchanged between the routers is untouched.

VACLs can be easily deployed to enforce the proper trust model. Let's analyze our DMZ case. Servers at the DMZ are supposed to serve incoming connections only, and they are not expected to initiate connections to the outside world. A VACL can be applied to their secondary VLAN in order to control the traffic leaving these servers. It is crucial to note that when using VACLs, the traffic is dropped in hardware so there is no impact on the CPU of the router nor of the switch. Even in the case that one of the servers is involved in a Distributed Denial of Service (DDoS) attack as a source, the switch will drop all illegitimate traffic at wire speed, without any performance penalty. Similar filters can be applied in the router or firewall where servers are connected to, but this usually has severe performance implications.

Known Limitations of VACLs and PVLANS

When configuring filtering with VACLs, you should be careful with regard to the fragment handling on the PFC, and that the configuration is tuned according to the specification of the hardware.

Given the hardware design of the PFC of the Supervisor 1 of the Catalyst 6500, it is better to explicitly deny

the icmp fragments. The reason is that Internet Control Message Protocol (ICMP) fragments and echo-reply are considered the same by the hardware, and by default the hardware is programmed to explicitly permit fragments. So if you want to stop echo-reply packets from leaving the servers, you explicitly have to configure this with the line **deny icmp any any fragment**. The configurations in this document take this into account.

There is a well-known security limitation to PVLANS, which is the possibility that a router forwards traffic back out of the same subnet from which it came. A router can route traffic across isolated ports defeating the purpose of PVLANS. This limitation is due to the fact that PVLANS are a tool that provides isolation at L2, not at Layer 3 (L3).

There is a fix to this problem, which is achieved by means of VACLs configured on the primary VLANs. The case study provides the VACLs that need to be configured on the primary VLAN to drop the traffic originated by the same subnet and routed back to the same subnet.

On some line cards, the configuration of PVLAN mappings / maps / trunking ports is subject to some restrictions where multiple PVLAN mappings have to belong to different port Application-Specific Integrated Circuits (ASICs) in order to get configured. Those restrictions are removed on the new port ASIC Coil3. Please refer to the latest Catalyst switch documentation on software configuration for these details.

Example Case Studies

The following section describes three case studies, which we believe are representative of most implementations and give the details related to the security deployment of PVLANS and VACLs.

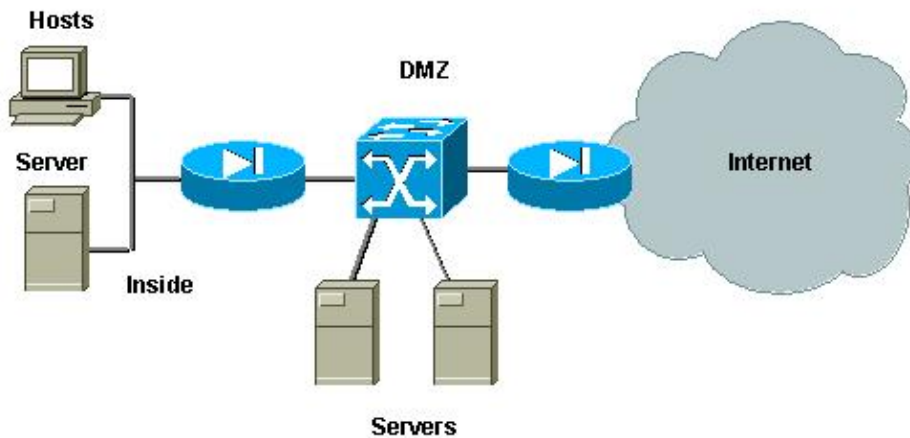
These scenarios are:

- Pass-Through DMZ
- External DMZ
- VPN Concentrator in Parallel to Firewall

Pass-Through DMZ

This is one of the most commonly deployed scenarios. In this example, the DMZ is implemented as a transit area between two firewall routers as illustrated in the image below.

Figure 2: Pass-Through DMZ



In this example, DMZ servers are supposed to be accessed by external as well as internal users, but they don't need to communicate with each other. In some cases, DMZ servers need to open some kind of connection to an internal host. At the same time, internal clients are supposed to access the Internet without restrictions. A good example will be the one with Web servers at the DMZ, which need to communicate with a database server located in the inside network, and having inside clients accessing the Internet.

The external firewall is configured to allow incoming connections to the servers located at the DMZ, but usually no filter or restrictions are applied to the outgoing traffic, particularly the traffic originated in the DMZ. As we discussed earlier in this document, this can potentially facilitate the activity of an attacker for two reasons: the first one, as soon as one of the DMZ hosts is compromised, all other DMZ hosts are exposed; the second one, an attacker can easily exploit an outgoing connection.

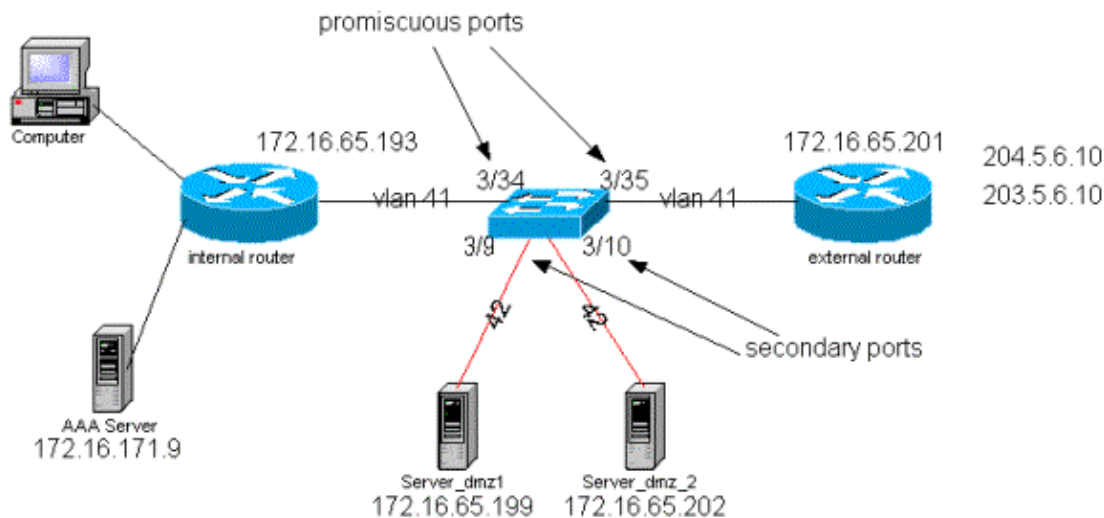
Since DMZ servers don't need to talk to each other, the recommendation is to make sure they are isolated at L2. The servers ports will be defined as PVLANS isolated ports, while the ports connecting to the two firewalls will be defined as promiscuous. Defining a primary VLAN for the firewalls, and a secondary VLAN for the DMZ servers will achieve this.

VACLs will be used to control the traffic originated in the DMZ. This will prevent an attacker from being able to open an illegitimate outgoing connection. It is important to keep in mind DMZ servers will not only need to reply with the traffic corresponding to client sessions, but they will also need some additional services, such as Domain Name System (DNS) and maximum transmission unit (MTU) path discovery. So, the ACL should allow all services needed by the DMZ servers.

Testing Pass-Through DMZ

In our test-bed we have implemented a DMZ segment with two routers configured as bed servers, server_dmz1 and server_dmz2. These servers are supposed to be accessed by outside as well as inside clients, and all HTTP connections are authenticated by using an internal RADIUS server (CiscoSecure ACS for UNIX). Both internal and external routers are configured as packet filter firewalls. The following picture illustrates the test-bed, including the addressing scheme used.

Figure 3: Pass-Through DMZ Test-Bed



The following list collects the fundamental configuration steps of PVLANS. The Catalyst 6500 is used as the L2 switch in the DMZ.

- Server_dmz_1 is connected to port 3/9
- Server_dmz_2 is connected to port 3/10
- The internal router is connected to port 3/34
- The external router is connected to port 3/35

We chose the following VLANs:

- 41 is the primary VLAN
- 42 is the isolated VLAN

Private VLAN Configuration

The following configuration sets the PVLANS on the ports involved.

```

ecomm-6500-2 (enable) set vlan 41 pvlan primary
VTP advertisements transmitting temporarily stopped,
and will resume after the command finishes.
Vlan 41 configuration successful

ecomm-6500-2 (enable) sh pvlan
Primary Secondary Secondary-Type Ports
-----
41 - -
ecomm-6500-2 (enable) set vlan 42 pvlan isolated
VTP advertisements transmitting temporarily stopped,
and will resume after the command finishes.
Vlan 42 configuration successful
ecomm-6500-2 (enable) set pvlan 41 42 3/9-10
Successfully set the following ports to Private Vlan 41,42:
3/9-10

ecomm-6500-2 (enable) set pvlan mapping 41 42 3/35
Successfully set mapping between 41 and 42 on 3/35
ecomm-6500-2 (enable) set pvlan mapping 41 42 3/34
Successfully set mapping between 41 and 42 on 3/34

```

Port	Name	Status	Vlan	Duplex	Speed	Type

```

3/9  server_dmz1          connected  41,42      a-half  a-10 10/100BaseTX
3/10 server_dmz2          connected  41,42      a-half  a-10 10/100BaseTX

3/34 to_6500_1          connected  41          auto   auto 10/100BaseTX
3/35 external_router_dm connected  41          a-half a-10 10/100BaseTX

```

VACL Configuration on the Primary VLAN

This section is crucial to improve security on the DMZ. As described in the Known Limitations of VACLs and PVLANS section, even if servers belong to two different secondary VLANs or to the same isolated VLAN, there is still a way an attacker can use to make them communicate to each other. If the servers try to communicate directly, they will not be able to do it at L2 because of the PVLANS. If the servers are compromised and then configured by an intruder in such a way that the traffic for the same subnet is sent to the router, this one will route the traffic back on the same subnet, thus defeating the purpose of the PVLANS.

Therefore, a VACL needs to be configured on the primary VLAN (the VLAN that carries the traffic from the routers) with the following policies:

- Allow the traffic whose source IP is the IP of the router
- Deny the traffic with both source and destination IPs being the DMZ subnet
- Allow all the rest of the traffic

```

ecomm-6500-2 (enable) sh sec acl info protect_pvlan
set security acl ip protect_pvlan
-----
1. permit ip host 172.16.65.193 any
2. permit ip host 172.16.65.201 any
3. deny ip 172.16.65.192 0.0.0.15 172.16.65.192 0.0.0.15
4. permit ip any any

ecomm-6500-2 (enable) sh sec acl
ACL                                     Type  VLANs
-----
protect_pvlan                          IP    41

```

This ACL will not affect the traffic generated by the servers; it will only prevent the routers from routing the traffic coming from the servers back to the same VLAN. The first two statements allow the routers to send messages such as icmp redirect or icmp unreachable to the servers.

VACL Configuration on the Secondary VLAN

The following configuration logs are used to show how we setup a VACL to filter the traffic generated by the servers. By configuring this VACL we want to achieve the following:

- Allow **ping** from servers (allow **echo**)
- Prevent **echo** replies from leaving the servers
- Allow HTTP connections originated from outside
- Allow RADIUS authentication (UDP port 1645) and accounting (UDP port 1646) traffic
- Allow DNS traffic (UDP port 53)

We want to prevent all the rest of the traffic.

As far as fragmentation is concerned, we assume the following on the server segment:

- The servers will not generate fragmented traffic

- The servers might receive fragmented traffic

Given the hardware design of the PFC of the Supervisor 1 of the Catalyst 6500, it is better to explicitly deny the icmp fragments. The reason is that ICMP fragments and echo-reply are considered the same by the hardware, and by default the hardware is programmed to explicitly permit fragments. So if you want to stop echo-reply packets from leaving the servers you explicitly have to configure this with the line **deny icmp any any fragment**.

```

ecomm-6500-2 (enable) Set sec acl ip dmz_servers_out deny icmp any any fragment
ecomm-6500-2 (enable) Set sec acl ip dmz_servers_out permit icmp host 172.16.65.199 any echo
ecomm-6500-2 (enable) Set sec acl ip dmz_servers_out permit icmp host 172.16.65.202 any echo
ecomm-6500-2 (enable) Set sec acl ip dmz_servers_out permit tcp host 172.16.65.199 eq 80 any established
ecomm-6500-2 (enable) Set sec acl ip dmz_servers_out permit tcp host 172.16.65.202 eq 80 any established
ecomm-6500-2 (enable) Set sec acl ip dmz_servers_out permit udp host 172.16.65.199
eq 1645 host 172.16.171.9 eq 1645
ecomm-6500-2 (enable) Set sec acl ip dmz_servers_out permit udp host 172.16.65.202
eq 1645 host 172.16.171.9 eq 1645
ecomm-6500-2 (enable) Set sec acl ip dmz_servers_out permit udp host 172.16.65.199
eq 1646 host 172.16.171.9 eq 1646
ecomm-6500-2 (enable) Set sec acl ip dmz_servers_out permit udp host 172.16.65.202
eq 1646 host 172.16.171.9 eq 1646
ecomm-6500-2 (enable) Set sec acl ip dmz_servers_out permit udp host 172.16.65.199 any eq 53
ecomm-6500-2 (enable) Set sec acl ip dmz_servers_out permit udp host 172.16.65.202 any eq 53

ecomm-6500-2 (enable) Commit sec acl all

ecomm-6500-2 (enable) Set sec acl map dmz_servers_out 42

ecomm-6500-2 (enable) sh sec acl
ACL                               Type  VLANs
-----
protect_pvlan                     IP    41
dmz_servers_out                   IP    42

ecomm-6500-2 (enable) sh sec acl info dmz_servers_out
set security acl ip dmz_servers_out
-----
1. deny icmp any any fragment
2. permit icmp host 172.16.65.199 any echo
3. permit icmp host 172.16.65.202 any echo
4. permit tcp host 172.16.65.199 eq 80 any established
5. permit tcp host 172.16.65.202 eq 80 any established
6. permit udp host 172.16.65.199 eq 1645 host 172.16.171.9 eq 1645
7. permit udp host 172.16.65.202 eq 1645 host 172.16.171.9 eq 1645
8. permit udp host 172.16.65.199 eq 1646 host 172.16.171.9 eq 1646
9. permit udp host 172.16.65.202 eq 1646 host 172.16.171.9 eq 1646
10. permit udp host 172.16.65.199 any eq 53
11. permit udp host 172.16.65.202 any eq 53

```

Testing the Configuration

The following output was captured when PVLANS were configured but no VACL were yet applied. This test is showing that from the external router the user is able to **ping** the internal router as well as the servers.

```

external_router#ping 172.16.65.193
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.65.193, timeout is 2 seconds:
!!!!

external_router#ping 172.16.65.202
Type escape sequence to abort.

```

```

Sending 5, 100-byte ICMP Echos to 172.16.65.202, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
external_router#ping 172.16.65.199
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.65.199, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

```

The following example shows that we are able to **ping** from the servers to the external network, the default gateway, but not the servers belonging to the same secondary VLAN.

```

server_dmz1#ping 203.5.6.10

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 203.5.6.10, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.65.193, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms

server_dmz1#ping 172.16.65.202
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.65.202, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

```

After mapping the VACLs, the **ping** from the external router is not going to succeed any more:

```

external_router#ping 172.16.65.199
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.65.199, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

```

The following example shows the server receiving HTTP GET requests from the internal network:

```

server_dmz1#debug ip http url
HTTP URL debugging is on
server_dmz1#debug ip http tran
HTTP transactions debugging is on
server_dmz1#debug ip http auth
HTTP Authentication debugging is on
server_dmz1#
*Mar 7 09:24:03.092 PST: HTTP: parsed uri '/'
*Mar 7 09:24:03.092 PST: HTTP: client version 1.0
*Mar 7 09:24:03.092 PST: HTTP: parsed extension Connection
*Mar 7 09:24:03.092 PST: HTTP: parsed line Keep-Alive
*Mar 7 09:24:03.092 PST: HTTP: parsed extension User-Agent
*Mar 7 09:24:03.092 PST: HTTP: parsed line Mozilla/4.7 [en] (X11; I; SunOS 5.5.1 sun4u)
*Mar 7 09:24:03.092 PST: HTTP: parsed extension Host
*Mar 7 09:24:03.092 PST: HTTP: parsed line 172.16.65.199
*Mar 7 09:24:03.092 PST: HTTP: parsed extension Accept
*Mar 7 09:24:03.092 PST: HTTP: parsed line image/gif, image/x-xbitmap, image/jpeg, image
*Mar 7 09:24:03.092 PST: HTTP: parsed extension Accept-Encoding
*Mar 7 09:24:03.092 PST: HTTP: parsed line gzip
*Mar 7 09:24:03.096 PST: HTTP: parsed extension Accept-Language
*Mar 7 09:24:03.096 PST: HTTP: parsed line en
*Mar 7 09:24:03.096 PST: HTTP: parsed extension Accept-Charset

```

```

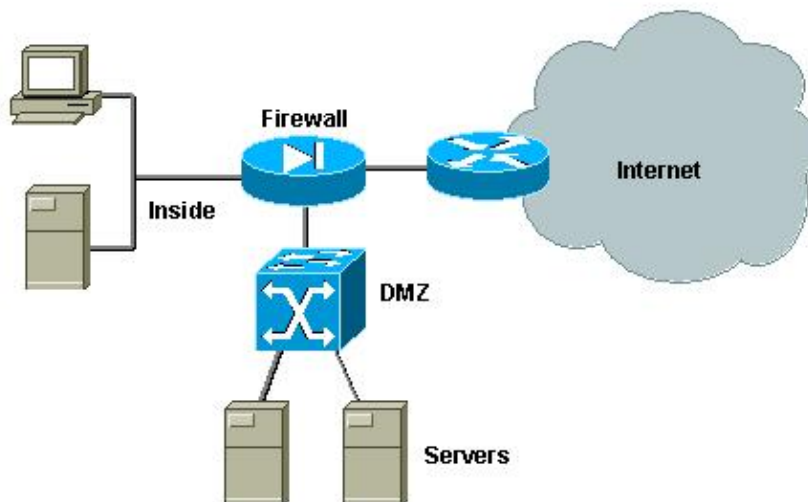
*Mar 7 09:24:03.096 PST: HTTP: parsed line iso-8859-1,*,utf-8
*Mar 7 09:24:03.096 PST: HTTP: Authentication for url '/' '/' level 15 privless '/'
*Mar 7 09:24:03.096 PST: HTTP: authentication required, no authentication information was
*Mar 7 09:24:03.096 PST: HTTP: authorization rejected
*Mar 7 09:24:22.528 PST: HTTP: parsed uri '/'
*Mar 7 09:24:22.532 PST: HTTP: client version 1.0
*Mar 7 09:24:22.532 PST: HTTP: parsed extension Connection
*Mar 7 09:24:22.532 PST: HTTP: parsed line Keep-Alive
*Mar 7 09:24:22.532 PST: HTTP: parsed extension User-Agent
*Mar 7 09:24:22.532 PST: HTTP: parsed line Mozilla/4.7 [en] (X11; I; SunOS 5.5.1 sun4u)
*Mar 7 09:24:22.532 PST: HTTP: parsed extension Host
*Mar 7 09:24:22.532 PST: HTTP: parsed line 172.16.65.199
*Mar 7 09:24:22.532 PST: HTTP: parsed extension Accept
*Mar 7 09:24:22.532 PST: HTTP: parsed line image/gif, image/x-xbitmap, image/jpeg, image
*Mar 7 09:24:22.532 PST: HTTP: parsed extension Accept-Encoding
*Mar 7 09:24:22.532 PST: HTTP: parsed line gzip
*Mar 7 09:24:22.532 PST: HTTP: parsed extension Accept-Language
*Mar 7 09:24:22.532 PST: HTTP: parsed line en
*Mar 7 09:24:22.532 PST: HTTP: parsed extension Accept-Charset
*Mar 7 09:24:22.532 PST: HTTP: parsed line iso-8859-1,*,utf-8
*Mar 7 09:24:22.532 PST: HTTP: parsed extension Authorization
*Mar 7 09:24:22.532 PST: HTTP: parsed authorization type Basic
*Mar 7 09:24:22.532 PST: HTTP: Authentication for url '/' '/' level 15 privless '/'
*Mar 7 09:24:22.532 PST: HTTP: Authentication username = 'martin' priv-level = 15 auth-ty
*Mar 7 09:24:22.904 PST: HTTP: received GET '

```

External DMZ

The external DMZ scenario is probably the most accepted and widely deployed implementation. An external DMZ is implemented by using one or more interfaces of a firewall, as shown the figure below.

Figure 4: External DMZ



Usually the requirements for DMZs tend to be the same regardless of the design implementation. As in the previous case, DMZ servers are supposed to be accessible from external clients as well as from the internal network. DMZ servers will eventually need access to some internal resources, and they are not supposed to talk to each other. At the same time, no traffic should be initiated from the DMZ to the Internet; these DMZ servers should only reply with traffic corresponding to incoming connections.

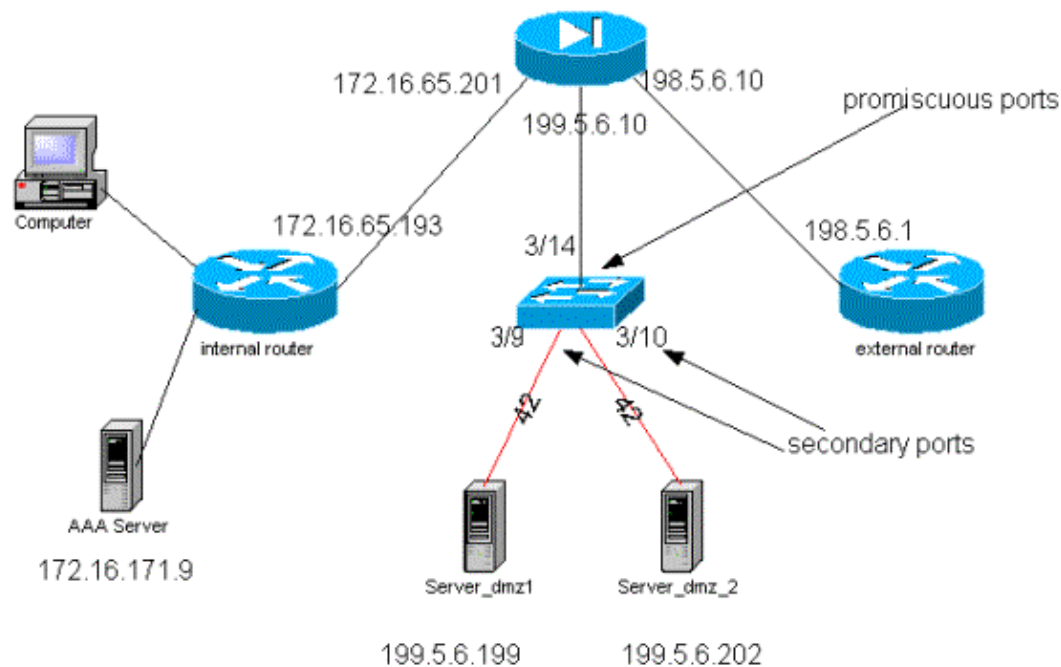
As in the previous case study, the first configuration step consists in achieving isolation at L2 by means of PVLANS, and to make sure the DMZ servers can't talk to each other while internal and external hosts can access them. This is implemented by setting the servers in a secondary VLAN with isolated ports. The firewall should be defined in a primary VLAN with a promiscuous port. The firewall will be the only device within this primary VLAN.

The second step is to define ACLs to control the traffic originated in the DMZ. When defining these ACLs we need to make sure only the necessary traffic is allowed.

Testing External DMZ

The image below shows the test-bed implemented for this case study, where we have used a PIX firewall with a third interface for the DMZ. The same set of routers is used as web servers, and all HTTP sessions are authenticated with the same RADIUS server.

Figure 5: External DMZ Test-Bed



For this scenario we attach only the more interesting excerpts from the configuration files, since the PVLANS and VACL configurations have been explained in detail in the previous case study.

PIX Configuration

```

nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 dmz security50
ip address outside 198.5.6.10 255.255.255.0
ip address inside 172.16.65.201 255.255.255.240
ip address dmz 199.5.6.10 255.255.255.0
global (outside) 1 198.5.6.11
global (dmz) 1 199.5.6.11
nat (inside) 1 0.0.0.0 0.0.0.0 0 0
static (dmz,outside) 199.5.6.199 199.5.6.199 netmask 255.255.255.255 0 0
static (dmz,outside) 199.5.6.202 199.5.6.202 netmask 255.255.255.255 0 0
static (inside,dmz) 172.16.171.9 172.16.171.9 netmask 255.255.255.255 0 0
static (inside,dmz) 171.68.10.70 171.68.10.70 netmask 255.255.255.255 0 0

```

```

static (inside,dmz) 171.69.0.0 171.69.0.0 netmask 255.255.0.0 0 0
conduit permit tcp host 199.5.6.199 eq www any
conduit permit tcp host 199.5.6.202 eq www any
conduit permit udp any eq domain any
conduit permit icmp any any echo-reply
conduit permit icmp any any unreachable
conduit permit udp host 172.16.171.9 eq 1645 host 199.5.6.199
conduit permit udp host 172.16.171.9 eq 1646 host 199.5.6.199
conduit permit udp host 172.16.171.9 eq 1645 host 199.5.6.202
conduit permit udp host 172.16.171.9 eq 1645 host 199.5.6.202
conduit permit icmp any host 199.5.6.199 echo
conduit permit icmp any host 199.5.6.202 echo
route outside 0.0.0.0 0.0.0.0 198.5.6.1 1
route inside 171.69.0.0 255.255.0.0 172.16.65.193 1
route inside 171.68.0.0 255.255.0.0 172.16.65.193 1
route inside 172.16.0.0 255.255.0.0 172.16.65.193 1

```

RADIUS Configuration

NAS Configuration

```

aaa new-model
aaa authentication login default radius local
aaa authentication login consoleauth none
aaa authorization exec default radius local
aaa authorization exec consoleautho none
aaa accounting exec default start-stop radius
aaa accounting exec consoleacct none
radius-server host 172.16.171.9 auth-port 1645 acct-port 1646
radius-server key cisco123
!
line con 0
  exec-timeout 0 0
  password ww
  authorization exec consoleautho
  accounting exec consoleacct
  login authentication consoleauth
  transport input none
line aux 0
line vty 0 4
  password ww
!
end

```

RADIUS Server CSUX

```

User Profile Information
user = martin{
  profile_id = 151
  profile_cycle = 5
  radius=Cisco {
    check_items= {
      2=cisco
    }
  }
  reply_attributes= {
    6=6
  }
}

User Profile Information
user = NAS.172.16.65.199{

```

```

profile_id = 83
profile_cycle = 2
NASName="172.16.65.199"
SharedSecret="cisco123"
RadiusVendor="Cisco"
Dictionary="DICTIONARY.Cisco"

}

```

Catalyst Configuration

It should be noticed that in this configuration there is no need to configure a VACL on the primary VLAN because the PIX does not redirect traffic out of the same interface it came from. A VACL as the one described in the VACL Configuration on the Primary VLAN section would be redundant.

```

set security acl ip dmz_servers_out
-----
1. deny icmp any any fragment
2. permit icmp host 199.5.6.199 any echo
3. permit icmp host 199.5.6.202 any echo
4. permit tcp host 199.5.6.199 eq 80 any established
5. permit tcp host 199.5.6.202 eq 80 any established
6. permit udp host 199.5.6.199 eq 1645 host 172.16.171.9 eq 1645
7. permit udp host 199.5.6.202 eq 1645 host 172.16.171.9 eq 1645
8. permit udp host 199.5.6.199 eq 1646 host 172.16.171.9 eq 1646
9. permit udp host 199.5.6.202 eq 1646 host 172.16.171.9 eq 1646
10. permit udp host 199.5.6.199 any eq 53
11. permit udp host 199.5.6.202 any eq 53
ecomm-6500-2 (enable) sh pvlan
Primary Secondary Secondary-Type Ports
-----
41 42 isolated 3/9-10

ecomm-6500-2 (enable) sh pvlan mapping
Port Primary Secondary
-----
3/14 41 42
3/34 41 42
3/35 41 42
ecomm-6500-2 (enable) sh port
Port Name Status Vlan Duplex Speed Type
-----
3/9 server_dmz1 connected 41,42 a-half a-10 10/100BaseTX
3/10 server_dmz2 connected 41,42 a-half a-10 10/100BaseTX
3/14 to_pix_port_2 connected 41 full 100 10/100BaseTX
3/35 external_router_dm notconnect 41 auto auto 10/100BaseTX

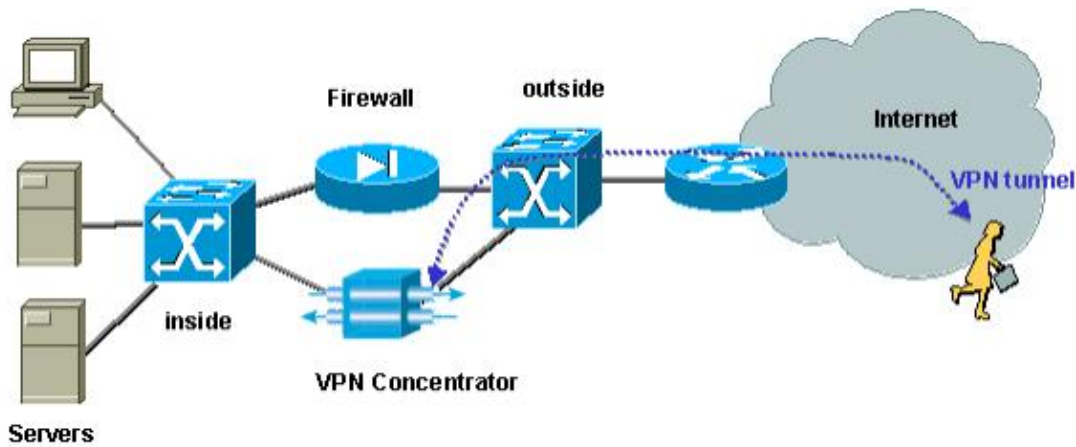
```

VPN Concentrator in Parallel to Firewall

When implementing Access Virtual Private Networks (VPNs), undoubtedly one of the favorite approaches is the parallel design (illustrated in the image below). Customers usually prefer this design approach since it is easy to implement, with almost no impact to the existing infrastructure, and because it is relatively easy to scale based on the device flexibility.

In the parallel approach, the VPN concentrator connects to both inside and outside segments. All VPN sessions terminate at the concentrator without going through the firewall. Usually VPN clients are expected to have unrestricted access to the inside network, but sometimes their access can be restricted to a set of inside servers (server farm). One of the desirable features is to segregate the VPN traffic from the regular Internet traffic, so for example, VPN clients are not allowed to access the Internet via the corporate firewall.

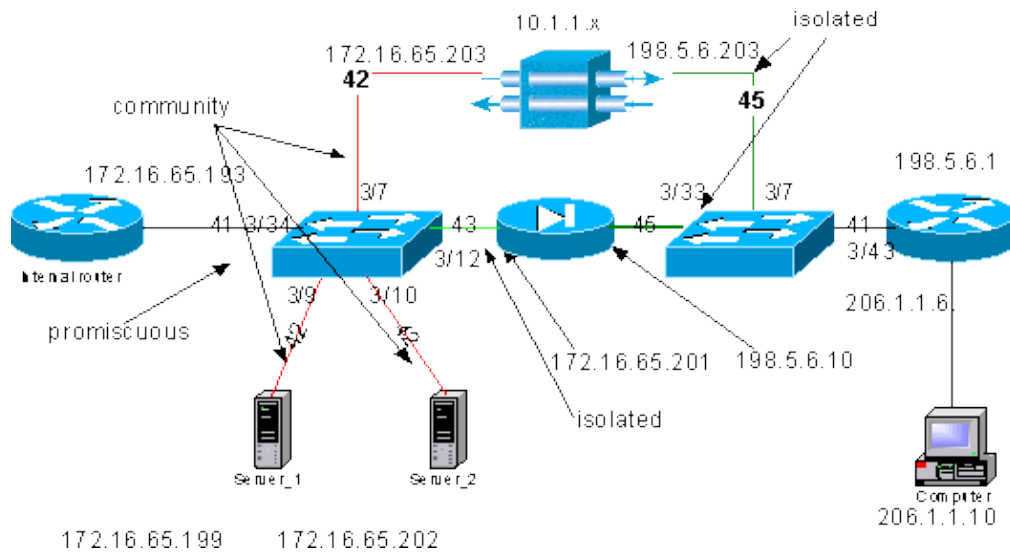
Figure 6: VPN Concentrator in Parallel to Firewall



Testing VPN Concentrator in Parallel to Firewall

In this example, we used a VPN 5000 Concentrator, which was installed in parallel to a PIX firewall. The two routers configured as Web servers were installed at the inside segment as an internal server farm. VPN clients are only allowed to access the server farm, and Internet traffic should be segregated from VPN traffic (IPSec). The figure below shows the test-bed.

Figure 7: VPN Concentrator in Parallel to Firewall Test-Bed



In this scenario we have two major areas of interest:

- The internal L2 switch
- The external L2 switch

The traffic flows for the internal L2 switch are defined based on the following statements:

- VPN clients have full access to a predefined set of internal servers (server farm)
- Internal clients are also allowed to access the server farm
- Internal clients have unrestricted access to the Internet
- Traffic coming from the VPN concentrator must be isolated from the PIX firewall

The traffic flows for the external L2 switch are defined as follows:

- Traffic coming from the router must be able to go either to the VPN concentrator or the PIX
- Traffic coming from the PIX must be isolated from the traffic coming from the VPN

Additionally it is possible that the administrator wants to prevent traffic from the internal network from being able to make its way to the VPN hosts, this can be achieved by means of VACLs configured on the primary VLAN (the VACL will filter only the traffic leaving from the internal router, no other traffic will be affected).

PVLAN Configuration

Since the main objective in this design is to keep the traffic coming from the PIX segregated from the traffic coming from the servers and from the VPN concentrator, we configure the PIX on a different PVLAN than the PVLAN on which the servers and the VPN concentrator are configured.

The traffic coming from the internal network must be able to access the server farm as well as the VPN concentrator and the PIX. As a consequence, the port that connects to the internal network is going to be a promiscuous port.

The servers and the VPN concentrator belong to the same secondary VLAN because they will be able to communicate with each other.

As for the external L2 switch, the router that gives access to the Internet (which typically belongs to an Internet Service Provider (ISP)) is connected to a promiscuous port while the VPN concentrator and the PIX belong to the same private and isolated VLANs (so that they cannot exchange any traffic). By doing this, the traffic coming from the service provider can take either the path to the VPN concentrator or the path to the PIX. The PIX and VPN concentrator are more protected since they are isolated.

PVLAN Configuration of the Internal L2 Switch

```
sh pvlan
Primary Secondary Secondary-Type Ports
-----
41      42      community      3/7,3/9-10
41      43      isolated       3/12

ecomm-6500-2 (enable) sh pvlan map
Port Primary Secondary
-----
3/34 41      42-43

ecomm-6500-2 (enable) sh port 3/7
Port Name Status Vlan Duplex Speed Type
-----
3/7 to_vpn_conc connected 41,42 a-half a-10 10/100BaseTX

ecomm-6500-2 (enable) sh port 3/9
Port Name Status Vlan Duplex Speed Type
-----
3/9 server_1 connected 41,42 a-half a-10 10/100BaseTX

ecomm-6500-2 (enable) sh port 3/10
Port Name Status Vlan Duplex Speed Type
-----
3/10 server_2 connected 41,42 a-half a-10 10/100BaseTX

ecomm-6500-2 (enable) sh port 3/12
```

Port	Name	Status	Vlan	Duplex	Speed	Type
3/12	to_pix_intf1	connected	41,43	a-full	a-100	10/100BaseTX

```
ecomm-6500-2 (enable) sh pvlan map
```

```
Port Primary Secondary
```

```
-----
3/34 41      42-43
```

```
ecomm-6500-2 (enable) sh port 3/34
```

Port	Name	Status	Vlan	Duplex	Speed	Type
3/34	to_int_router	connected	41	a-full	a-100	10/100BaseTX

PVLAN Configuration of the External L2 Switch

```
sh pvlan
```

Primary	Secondary	Secondary-Type	Ports
41	45	isolated	3/7,3/33

```
ecomm-6500-1 (enable) sh pvlan mapping
```

```
Port Primary Secondary
```

```
-----
3/43 41      45
```

```
ecomm-6500-1 (enable) sh port 3/7
```

Port	Name	Status	Vlan	Duplex	Speed	Type
3/7	from_vpn	connected	41,45	a-half	a-10	10/100BaseTX

```
ecomm-6500-1 (enable) sh port 3/33
```

Port	Name	Status	Vlan	Duplex	Speed	Type
3/33	to_pix_intf0	connected	41,45	a-full	a-100	10/100BaseTX

```
ecomm-6500-1 (enable) sh pvlan map
```

```
Port Primary Secondary
```

```
-----
3/43 41      45
```

```
ecomm-6500-1 (enable) sh port 3/43
```

Port	Name	Status	Vlan	Duplex	Speed	Type
3/43	to_external_router	connected	41	a-half	a-10	10/100BaseTX

Testing the Configuration

This experiment shows that the internal router can go through the firewall and reach the external router (the external firewall router whose interface is 198.5.6.1).

```
ping 198.5.6.1
```

```
Type escape sequence to abort
```

```
Sending 5, 100-byte ICMP Echos to 198.5.6.1, timeout is 2 seconds:
```

```
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

This experiment shows the following, all from server 1:

- Server 1 can ping the internal router:

```
server_1#ping 172.16.65.193
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 172.16.65.193, timeout is 2 seconds:
```

```
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

- Server 1 can ping the VPN:

```
server_1#ping 172.16.65.203
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 172.16.65.203, timeout is 2 seconds:
```

```
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

- Server 1 cannot ping PIX internal interface:

```
server_1#ping 172.16.65.201
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 172.16.65.201, timeout is 2 seconds:
```

```
.....
```

```
Success rate is 0 percent (0/5)
```

- Server 1 cannot ping the external router:

```
server_1#ping 198.5.6.1
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 198.5.6.1, timeout is 2 seconds:
```

```
.....
```

```
Success rate is 0 percent (0/5)
```

The following experiment shows that HTTP sessions can be opened from the internal network to the server farm.

```
server_2#
```

```
lwld: HTTP: parsed uri '/'
```

```
lwld: HTTP: processing URL '/' from host 171.68.173.3
```

```
lwld: HTTP: client version 1.0
```

```
lwld: HTTP: parsed extension Connection
```

```
lwld: HTTP: parsed line Keep-Alive
```

```
lwld: HTTP: parsed extension User-Agent
```

```
lwld: HTTP: parsed line Mozilla/4.7 [en] (X11; I; SunOS 5.5.1 sun4u)
```

```
lwld: HTTP: parsed extension Host
```

```
lwld: HTTP: parsed line 172.16.65.202
```

```
lwld: HTTP: parsed extension Accept
```

```
lwld: HTTP: parsed line image/gif, image/x-xbitmap, image/jpeg, image/
```

```
lwld: HTTP: parsed extension Accept-Encoding
```

```
lwld: HTTP: parsed line gzip
```

```
lwld: HTTP: parsed extension Accept-Language
```

```
lwld: HTTP: parsed line en
```

```
lwld: HTTP: parsed extension Accept-Charset
```

```
lwld: HTTP: parsed line iso-8859-1,*,utf-8
```

```
lwld: HTTP: Authentication for url '/' '/' level 15 privless '/'
```

```
lwld: HTTP: authentication required, no authentication information was provided
```

```
lwld: HTTP: authorization rejected
```

```
lwld: HTTP: parsed uri '/'
```

```
lwld: HTTP: processing URL '/' from host 171.68.173.3
```

```
lwld: HTTP: client version 1.0
```

```
lwld: HTTP: parsed extension Connection
```

```
lwld: HTTP: parsed line Keep-Alive
```

```
lwld: HTTP: parsed extension User-Agent
```

```
lwld: HTTP: parsed line Mozilla/4.7 [en] (X11; I; SunOS 5.5.1 sun4u)
```

```

lwld: HTTP: parsed extension Host
lwld: HTTP: parsed line 172.16.65.202
lwld: HTTP: parsed extension Accept
lwld: HTTP: parsed line image/gif, image/x-xbitmap, image/jpeg, image/
lwld: HTTP: parsed extension Accept-Encoding
lwld: HTTP: parsed line gzip
lwld: HTTP: parsed extension Accept-Language
lwld: HTTP: parsed line en
lwld: HTTP: parsed extension Accept-Charset
lwld: HTTP: parsed line iso-8859-1,*,utf-8
lwld: HTTP: parsed extension Authorization
lwld: HTTP: parsed authorization type Basic
lwld: HTTP: Authentication for url '/' '/' level 15 privless '/'
lwld: HTTP: Authentication username = 'maurizio' priv-level = 15 auth-type = aaa
lwld: HTTP: received GET ''

```

The following experiment shows that the HTTP traffic from the VPN network can make its way to the server farm (notice the address 10.1.1.1).

```

lwld: HTTP: parsed uri '/'
lwld: HTTP: processing URL '/' from host 10.1.1.1
lwld: HTTP: client version 1.0
lwld: HTTP: parsed extension Connection
lwld: HTTP: parsed line Keep-Alive
lwld: HTTP: parsed extension User-Agent
lwld: HTTP: parsed line Mozilla/4.76 [en] (Windows NT 5.0; U)
lwld: HTTP: parsed extension Host
lwld: HTTP: parsed line 172.16.65.202
lwld: HTTP: parsed extension Accept\
lwld: HTTP: parsed line image/gif, image/x-xbitmap, image/jpeg, image/
lwld: HTTP: parsed extension Accept-Encoding
lwld: HTTP: parsed line gzip
lwld: HTTP: parsed extension Accept-Language
lwld: HTTP: parsed line en
lwld: HTTP: parsed extension Accept-Charset
lwld: HTTP: parsed line iso-8859-1,*,utf-8
lwld: HTTP: Authentication for url '/' '/' level 15 privless '/'
lwld: HTTP: authentication required, no authentication information was provided

```

The following is the configuration of the VPN concentrator:

```

[ IP Ethernet 0:0 ]
ipbroadcast = 172.16.65.255
mode = routedSubnetMask = 255.255.255.240
IPAddress = 172.16.65.203

[ General ]
IPsecGateway = 198.5.6.1
DeviceName = "VPN5008"
EnablePassword = "ww"
Password = "ww"
EthernetAddress = 00:30:85:14:5c:40
DeviceType = VPN 5002/8
ConcentratorConfiguredOn = Timeserver not configured
ConfiguredFrom = Command Line, from 171.68.173.3

[ IP Static ]
206.1.1.0 255.255.255.0
198.5.6.1 10.0.0.0
0.0.0.0 172.16.65.193 1

[ IP Ethernet 1:0 ]
ipbroadcast = 172.16.65.255

```

```

mode = routedSubnetMask          = 255.255.255.0
IPAddress                        = 198.5.6.203

[ IKE Policy ]
Protection = MD5_DES_G1

[ VPN Group "RemoteUsers" ]
maxconnections = 10IPNet          = 172.16.65.0/24
LocalIPNet     = 10.1.1.0/24
Transform = esp(des,md5)

[ VPN Users ]
martin Config="RemoteUsers"
SharedKey="mysecretkey"
maurizio Config="RemoteUsers"
SharedKey="mysecretkey"

```

The following command shows the list of users connected:

```

sh VPN user

```

Port	User	Group	Client Address	Local Address	ConnectNumber Time
VPN 0:1	martin	RemoteUsers	206.1.1.10	10.1.1.1	00:00:11:40

It should be noticed that the default gateway on the servers is the internal router 172.16.65.193, which will issue an icmp redirect to 172.16.65.203. This implementation causes non-optimal traffic flows, because the host would send the first packet of a flow to the router, and upon reception of the redirect, it will send the subsequent packets to the gateway that is more appropriate to handle this traffic. Alternatively one could configure two different routes on the servers themselves in order to point to the VPN for the 10.x.x.x addresses and to 172.16.65.193 for the rest of the traffic. If only the default gateway is configured on the servers, then we need to make sure that the router interface is configured with "ip redirect."

An interesting point that we noticed during the testing is the following one. If we try to **ping** an external address like 198.5.6.1 from the servers or from the VPN, the default gateway will send an icmp redirect to 172.16.65.201.

```

Sending 5, 100-byte ICMP Echos to 198.5.6.1, timeout is 2 seconds:
lwd: ICMP: redirect rcvd from 172.16.65.193 -- for 198.5.6.1 use gw 172.16.65.201.
lwd: ICMP: redirect rcvd from 172.16.65.193 -- for 198.5.6.1 use gw 172.16.65.201.
lwd: ICMP: redirect rcvd from 172.16.65.193 -- for 198.5.6.1 use gw 172.16.65.201.
lwd: ICMP: redirect rcvd from 172.16.65.193 -- for 198.5.6.1 use gw 172.16.65.201.
lwd: ICMP: redirect rcvd from 172.16.65.193 -- for 198.5.6.1 use gw 172.16.65.201.
Success rate is 0 percent (0/5)

```

The servers or the VPN will at this point send an Address Resolution Protocol (ARP) request for 172.16.65.201 and will not get any response back from 201 because it is on another secondary VLAN; this is what the PVLAN provides us. In reality there is an easy way to get round this, which is to send traffic to the MAC of .193 and with the destination IP of 172.16.65.201.

The router .193 will route the traffic back to the same interface, but since the router interface is a promiscuous port, the traffic will reach .201, which we wanted to prevent. This issue was explained in the Known Limitations of VACLs and PVLANS section.

VACL Configuration

This section is crucial to improve security on the server farm. As described in the Known Limitations of VACLs and PVLANS section, even if servers and the PIX belong to two different secondary VLANs, there is still a method an attacker can use to make them communicate to each other. If they try to communicate directly, they will not be able to do it because of the PVLANS. If the servers are compromised and then configured by an intruder in such a way that the traffic for the same subnet is sent to the router, this one will route the traffic back on the same subnet, thus defeating the purpose of the PVLANS.

Therefore, A VACL needs to be configured on the primary VLAN (the VLAN that carries the traffic from the routers) with the following policies:

- Allow the traffic whose source IP is the IP of the router
- Deny the traffic with both source and destination IPs being the server farm's subnet
- Allow all the rest of the traffic

```
ecomm-6500-2 (enable) sh sec acl info protect_pvlan
set security acl ip protect_pvlan
-----
1. permit ip host 172.16.65.193 any
2. deny ip 172.16.65.192 0.0.0.15 172.16.65.192 0.0.0.15
3. permit ip any any

ecomm-6500-2 (enable) sh sec acl
ACL                               Type  VLANS
-----
protect_pvlan                     IP    41
```

This ACL will not affect the traffic generated by the servers nor by the PIX; it will only prevent the routers from routing the traffic coming from the servers back to the same VLAN. The first two statements allow the routers to send messages like icmp redirect or icmp unreachable to the servers.

We identified another traffic flow that the administrator might want to stop by means of VACLs, and this flow is from the internal network to the VPN hosts. In order to do so, a VACL can be mapped to the primary VLAN (41) and combined with the previous one:

```
show sec acl info all

set security acl ip protect_pvlan

1. deny ip any 10.1.1.0 0.0.0.255
2. permit ip host 172.16.65.193 any
3. deny ip 172.16.65.192 0.0.0.15 172.16.65.192 0.0.0.15
4. permit ip any any
```

Testing the Configuration

We are now pinging the 10.1.1.1 host from the router .193 (zundapp). Before we map the VACL, the **ping** is successful.

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

After mapping the VACL on VLAN 41, the same **ping** will not succeed:

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:  
.....  
Success rate is 0 percent (0/5)
```

However, we can still **ping** the external router:

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 198.5.6.1, timeout is 2 seconds:  
!!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 100/171/192 ms
```

Related Information

- [Configuring Access Control Lists – Catalyst 6000 Documentation](#)
- [Technical Support – Cisco Systems](#)

All contents are Copyright © 1992–2005 Cisco Systems, Inc. All rights reserved. Important Notices and Privacy Statement.

Updated: Mar 15, 2005

Document ID: 10601
