

PIX/ASA 7.x and IOS: VPN Fragmentation

Document ID: 82444

Introduction

Prerequisites

- Requirements
- Components Used
- Network Diagram
- Related Products
- Conventions

Background Information

- Issues with Fragmentation

Main Task

- Discover Fragmentation
- Solutions to Fragmentation Issues

Verify

Troubleshoot

- VPN Encryption Error
- RDP and Citrix Problems

Related Information

Introduction

This document walks you through the steps required to alleviate problems that can occur with the fragmentation of a packet. An example of a fragmentation problem is the ability to ping a networked resource but the inability to connect to that same resource with a specific application, such as E-mail or databases.

Prerequisites

Requirements

Ensure that you meet these requirements before you attempt this configuration:

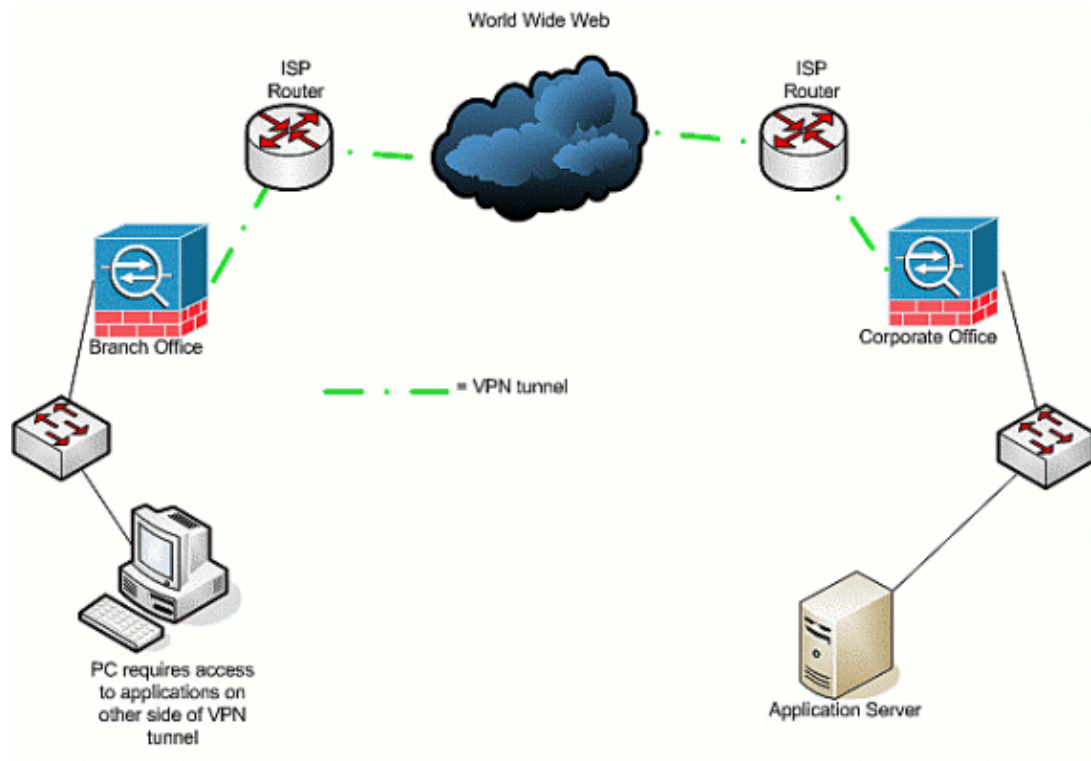
- Connectivity between VPN peers

Components Used

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

Network Diagram

This document uses this network setup:



Related Products

This configuration can also be used with these hardware and software versions:

- IOS Routers
- PIX/ASA security devices

Conventions

Refer to the Cisco Technical Tips Conventions for more information on document conventions.

Background Information

IP supports a maximum length of 65,536 bytes for an IP packet, but most data-link layer protocols support a much smaller length, called a maximum transmission unit (MTU). Based on the supported MTU, it can be necessary to break up (fragment) an IP packet to transmit it across a particular data-link layer media type. The destination then has to reassemble the fragments back into the original, complete IP packet.

Protocol	Additional Bytes
ESP (encryption and hash)	56
AH	24+
GRE	24
NAT-T/IPsec over UDP (UDP part)	8
IPsec over TCP (TCP part)	20
L2TP	12
PPTP	48
Outer IP header in IPsec tunnel mode or PPTP/L2TP	20
PPPoE	8

When you use a VPN to protect data between two VPN peers, additional overhead is added to the original data, which can require that fragmentation occur. This table lists fields that potentially have to be added to the protected data in order to support a VPN connection. Note that multiple protocols can be necessary, which increases the size of the original packet. For example, if you use a L2L DMVPN IPSEC connection between two Cisco routers, where you have implemented a GRE tunnel, you need this additional overhead: ESP, GRE, and the outer IP header. If you have an IPsec software client connection to a VPN gateway when the traffic goes through an address device, you need this additional overhead for Network Address Translation–Traversal (NAT–T), as well as the outer IP header for the tunnel mode connection.

Issues with Fragmentation

When the source sends a packet to a destination, it places a value in the control flags field of the IP headers that affects fragmentation of the packet by intermediate devices. The control flag is three bits long, but only the first two are used in fragmentation. If the second bit is set to 0, the packet is allowed to be fragmented; if it is set to 1, the packet is not allowed to be fragmented. The second bit is commonly called the *don't fragment* (DF) bit. The third bit specifies when the fragmentation occurs, whether or not this fragmented packet is the last fragment (set to 0), or if there are more fragments (set to 1) that make up the packet.

There are four areas that can create problems when fragmentation is required:

- Additional overhead in CPU cycles and memory is required by the two devices that perform fragmentation and reassembly.
- If one fragment is dropped on the way to the destination, the packet cannot be reassembled and the entire packet must be fragmented and sent again. This creates additional throughput problems, especially in situations where the traffic in question is rate–limited, and the source sends traffic above the allowable limit.
- Packet filtering and stateful firewalls can have difficulty processing the fragments. When fragmentation occurs, the first fragment contains an outer IP header, the inner header, such as TCP, UDP, ESP and others, and part of the payload. Subsequent fragments of the original packet contain an outer IP header and the continuation of the payload. The problem with this process is that certain firewalls need to see the inner header information in every packet in order to make intelligent filtering decisions; if that information is missing, they inadvertently can drop all fragments, except for the first one.
- The source in the IP header of the packet can set the third control bit to *don't fragment*, which means that, if an intermediate device receives the packet and must fragment it, the intermediate device cannot fragment it. Instead, the intermediate device drops the packet.

Main Task

Discover Fragmentation

Most networks use Ethernet, with a default MTU value of 1,500 bytes, that is typically used for IP packets. In order to find out if fragmentation occurs or is needed but cannot be done (DF bit is set), first bring your VPN session up. Then you can use any one of these four procedures to discover fragmentation.

1. Ping a device located at the other end. This is under the assumption that pinging is allowed across the tunnel. If this is successful, try to access an application across the same device; for example, if a Microsoft E-mail or Remote Desktop server is across the tunnel, open Outlook and try to download your E-mail, or try to Remote Desktop to the server. If this does not work, and you have the correct name resolution, there is a good chance that fragmentation is the issue.
2. From a Windows device use this: C:\> **ping -f -l packet_size_in_bytes destination_IP_address.**

The **-f** option is used to specify that the packet cannot be fragmented. The **-l** option is used to specify the length of the packet. First try this with a packet size of 1,500. For example, ping **-f -l 1500 192.168.100**. If fragmentation is required but cannot be performed, you receive a message such as this: *Packets need to be fragmented but DF set.*

3. On Cisco routers, execute the **debug ip icmp** command and use the **extended ping** command. If you see *ICMP:dst (x.x.x.x) fragmentation needed and DF set, unreachable sent to y.y.y.y*, where x.x.x.x is a destination device, and y.y.y.y is your router, an intermediate device tells you that fragmentation is needed, but because you set the DF bit in the echo request, an intermediate device cannot fragment it in order to forward it to the next hop. In this case, gradually decrease the MTU size of the pings until you find one that works.
4. On Cisco Security Appliances, use a capture filter.

```
◆ ciscoasa(config)#access-list outside_test permit tcp any host 172.22.1.1 eq 80
```

Note: When you leave the source as *any*, it allows the administrator to monitor any network address translations (NAT).

```
◆ ciscoasa(config)#access-list outside_test permit tcp host 172.22.1.1 eq 80 any
```

Note: When you reverse the source and destination information, it allows return traffic to be captured.

```
◆ ciscoasa(config)# capture outside_interface access-list outside_test interface outside
```

The user needs to initiate a new session with application X. After the user has initiated a new application X session, the ASA administrator needs to issue the **show capture outside_interface** command.

Solutions to Framentation Issues

There are different ways that you can solve issues with fragmentation. These are discussed in this section.

Method 1: Static MTU Setting

The static MTU setting can solve issues with fragmentation.

1. **MTU Change on the Router:**

Note that if you manually set the MTU on the device, it tells the device, which acts as a VPN gateway, to fragment received packets before it protects and sends them across the tunnel. This is preferable to having the router protect the traffic and then fragment it, but the device fragments it.



Warning: If you change the MTU size on any device interface, it causes all tunnels terminated on that interface to be torn down and rebuilt.

On Cisco routers, use the **ip mtu** command to adjust the MTU size on the interface where the VPN is terminated:

```
router (config)# interface type [slot_#/] port_#  
  
router (config-if)# ip mtu MTU_size_in_bytes
```

2. MTU Change on the ASA/PIX:

On ASA/PIX devices, use the **mtu** command to adjust the MTU size in global config mode. By default, the MTU is set to 1500. For example, if you had an interface on your security appliance that was named *Outside* (where the VPN is terminated), and you determined (through the measures listed in the Discover Fragmentation section) that you wanted to use 1380 as the fragment size, use this command:

```
security appliance (config)# mtu Outside 1380
```

Method 2: TCP Maximum Segment Size

The TCP maximum segment size can solve issues with fragmentation.

Note: This feature only works with TCP; other IP protocols have to use another solution to solve IP fragmentation problems. Even if you set the **ip mtu** on the router, it does not affect what the two end hosts negotiate within the TCP three-way handshake with TCP MSS.

1. MSS Change on the Router:

Fragmentation occurs with TCP traffic because TCP traffic is normally used to transport large amounts of data. TCP supports a feature called TCP maximum segment size (MSS) that allows the two devices to negotiate a suitable size for TCP traffic. The MSS value is configured statically on each device and represents the buffer size to use for an expected packet. When two devices establish TCP connections they compare the local MSS value with the local MTU value within the three-way handshake; whichever is lower is sent to the remote peer. The two peers then use the lower of the two exchanged values. In order to configure this feature, do this:

On Cisco routers, use the **tcp adjust-mss** command on the interface on which the VPN is terminated.

```
router (config)# interface type [slot_#/] port_#  
  
router (config-if)# ip tcp adjust-mss MSS_Size_in_bytes
```

2. MSS Change on the ASA/PIX:

In order to ensure that the maximum TCP segment size does not exceed the value you set and that the maximum is not less than a specified size, use the **sysopt connection** command in global config mode. In order to restore the default setting, use the **no** form of this command. The default maximum value is 1380 bytes. The minimum feature is disabled by default (set to 0).

In order to change the default maximum MSS limit, do this:

```
security appliance (config)# sysopt connection tcp-mss MSS_size_in_bytes
```

Note: If you set the maximum size to be greater than 1380, packets can become fragmented, dependent upon the MTU size (which is 1500 by default). Large numbers of fragments can impact the performance of the security appliance when it uses the Frag Guard feature. If you set the minimum size, it prevents the TCP server from sending many small TCP data packets to the client and impacting the performance of the server and the network.

In order to change the minimum MSS limit, do this:

```
security appliance (config)# sysopt connection tcp-mss minimum MSS_size_in_bytes  
  
security appliance (config)# sysopt connection tcp-mss minimum  
MSS_size_in_bytes
```

Note: Refer to the MPF Configuration to Allow Packets that Exceed MSS section of the document PIX/ASA 7.X Issue: MSS Exceeded – HTTP Clients Cannot Browse to Some Web Sites for more information in order to allow the exceeded MSS packets another method.

Method 3: Path MTU Discovery (PMTUD)

PMTUD can solve issues with fragmentation.

The main problem with TCP MSS is that the administrator has to know what value to configure on your router to prevent the occurrence of fragmentation. This can be a problem if more than one path exists between you and the remote VPN location, or, when you do your initial query, you find that the second–or third–smaller MTU, instead of the smallest, is based on the routing decision used within your initial query. With PMTUD, you can determine an MTU value for IP packets that avoids fragmentation. If ICMP messages are blocked by a router, the path MTU is broken, and packets with the DF bit set are discarded. Use the **set ip df** command to clear the DF bit and allow the packet to be fragmented and sent. Fragmentation can slow the speed of packet forwarding on the network, but access lists can be used to limit the number of packets on which the DF bit is cleared.

1. Three issues can cause PMTUD not to function:

- ◆ An intermediate router can drop the packet and not respond with an ICMP message. This is not very common on the Internet, but can be common inside a network where routers are configured to not respond with ICMP unreachable messages.
- ◆ An intermediate router can respond with an ICMP unreachable message, but, on the return flow, a firewall blocks this message. This is a more common occurrence.
- ◆ The ICMP unreachable message makes its way back to the source, but the source ignores the fragmentation message. This is the most uncommon of the three issues.

If you experience the first issue, you could either clear the DF bit in the IP header that the source placed there or manually adjust the TCP MSS size. In order to clear the DF bit, an intermediate router has to change the value from 1 to 0. Normally this is done by a router in your network before the packet leaves the network. This is a simple code configuration that does this on an IOS–based router:

```
Router (config) # access-list ACL_# permit tcp any any  
Router (config) # route-map route_map_name permit seq#  
  
Router (config-route-map) # match ip address ACL_#  
  
Router (config-route-map) # set ip df 0
```

```
Router (config-route-map) # exit
Router (config) # interface type [slot#/]port #

Router (config-if) # ip policy router-map route_map_name
```

2. PMTUD and GRE Tunnels

- ◆ By default, a router does not perform PMTUD on GRE tunnel packets that it generates itself. In order to enable PMTUD on GRE tunnel interfaces and have the router participate in the MTU tuning process for source/destination devices for traffic that traverses the tunnel, use this configuration:

```
◇ Router (config) # interface tunnel tunnel_#
◇ Router (config-if) # tunnel path-mtu-discovery
```

The **tunnel path-mtu-discovery** command enables PMTUD for the GRE tunnel interface of a router. The optional **age-timer** parameter specifies the number of minutes after which the tunnel interface resets the maximum MTU size discovered, minus 24 bytes for the GRE header. If you specify *infinite* for the timer, the timer is not used. The **min-mtu** parameter specifies the minimum number of bytes that comprises the MTU value.

3. PIX/ASA 7.x – Clear Don't Fragment (DF) or handling large files or packets.

You are still unable to properly access the Internet, large files, or applications through the tunnel because it gives this MTU size-error message:

```
PMTU-D packet 1440 bytes greater than effective mtu 1434,
dest_addr=10.70.25.1, src_addr=10.10.97.55, prot=TCP
```

In order to resolve this, be sure to clear the DF bit from the outside interface of the device. Configure the DF-bit policy for IPsec packets with the **crypto ipsec df-bit** command in global configuration mode.

```
pix(config)# crypto ipsec df-bit clear-df outside
```

The DF bit with IPsec tunnels feature lets you specify whether the security appliance can clear, set, or copy the Don't Fragment (DF) bit from the encapsulated header. The DF bit within the IP header determines whether a device is allowed to fragment a packet.

Use the **crypto ipsec df-bit** command in global configuration mode to configure the security appliance to specify the DF bit in an encapsulated header.

When you encapsulate tunnel mode IPsec traffic, use the **clear-df** setting for the DF bit. This setting lets the device send packets larger than the available MTU size. Also this setting is appropriate if you do not know the available MTU size.

Note: If you still experience fragmentation issues and dropped packets, optionally, you can manually adjust the MTU size with the **ip mtu tunnel interface** command. In this case, the router fragments the packet before it protects it. This command can be used in conjunction with PMTUD and/or TCP MSS.

Verify

There is currently no verification procedure available for this configuration.

The Output Interpreter Tool (registered customers only) (OIT) supports certain **show** commands. Use the OIT to view an analysis of **show** command output.

Troubleshoot

VPN Encryption Error

Assume that the IPSec Tunnel has established between the router and PIX. If you see encryption error messages that packets are dropped, complete these steps to resolve the issue:

1. Perform a sniffer trace from the client to the server side to find out which is the best MTU to use.

You can also use the ping test:

```
ping -l 1400 192.168.1.1 -f
```

192.168.1.1 is the IP address of the remote machine.

2. Continue to reduce the value of 1400 by 20 until there is a reply.

Note: The magical value, which works in most instances, is 1300.

3. After the appropriate maximum segment size is achieved, adjust it appropriately for the devices in use:

On the PIX Firewall:

```
sysopt connection tcpmss 1300
```

On the router:

```
ip tcp adjust-mss 1300
```

RDP and Citrix Problems

Problem:

You can ping between the VPN networks, but Remote Desktop Protocol (RDP) and Citrix connections cannot be established across the tunnel.

Solution:

The problem can be the MTU size on the PC behind the PIX/ASA. Set the MTU size as 1300 for the client machine and try to establish the Citrix connection across the VPN tunnel.

Related Information

- [Resolve IP Fragmentation, MTU, MSS, and PMTUD Issues with GRE and IPSEC](#)
 - [PIX/ASA 7.0 Issue: MSS Exceeded – HTTP Clients Cannot Browse to Some Web Sites](#)
 - [Most Common L2L and Remote Access IPsec VPN Troubleshooting Solutions](#)
 - [Why Can't I Browse the Internet when Using a GRE Tunnel](#)
 - [Technical Support & Documentation – Cisco Systems](#)
-

