

How NAT Handles ICMP Fragments

Document ID: 13771

Introduction

Prerequisites

Requirements

Components Used

Conventions

Case 1

Case 2

Case 3

Summary

NetPro Discussion Forums – Featured Conversations

Related Information

Introduction

This document explains how Network Address Translation (NAT) handles Internet Control Message Protocol (ICMP) fragments when you configure NAT overloading. For information about NAT overloading, refer to the NAT FAQ.

The handling of ICMP fragments depends on the state of the NAT translation table, and the order in which the NAT router receives the ICMP fragments. We'll look at three different cases, in which we send two pings from 172.16.0.1 to 172.17.1.2 with a length of 3600 bytes each (three IP fragments).

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

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

Conventions

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

Case 1

In this scenario we see NAT create a fully extended translation entry in the translation table. Once that's done, and there aren't any other usable addresses in the NAT pool, NAT drops any fragments received before the first fragment (fragment 0) of a packet.

As we start, only one address in the pool performs overload; the NAT translation table is empty; and the NAT configuration appears as:

```
ip nat pool POOL1 10.10.10.3 10.10.10.3 prefix-length 24
ip nat inside source list 5 pool POOL1 overload
```

```
access-list 5 permit 172.16.0.0 0.0.0.31
```

Let us look at what happens as packets start arriving at the NAT router.

1. Packet 1 fragment 0 arrives, and NAT creates a fully extended translation entry. NAT then translates and forwards packet 1 fragment 0. The translation table now appears as:

Pro	Inside global	Inside local	Outside local	Outside global
icmp	10.10.10.3:24320	172.16.0.1:24320	172.17.1.2:24320	172.17.1.2:24320

Note the number 24320 in the translation table above. It is the ICMP ident value included in the ICMP header of the IP datagram. Only fragment 0 of the IP datagram contains this ICMP header. To determine if multiple fragments are part of the same packet, NAT needs to track the IP ident value, found in the IP header of all fragments from the original IP datagram. If several fragments have the same IP ident value as fragment 0, which created the extended translation, NAT translates these fragments using the same extended translation entry. Refer to RFC 791 for more information about the IP identification field. Refer to RFC 792 for more information about the ICMP identification field.

2. Packet 1 fragment 2 and packet 1 fragment 1 arrive. Since these fragments are part of the same packet that contains fragment 0 (which created the translation), NAT uses the above translation entry to translate and forward these fragments. The destination device receives all fragments for packet 1 and sends a reply.
3. Packet 2 fragment 1 arrives. Since this is a new packet, its IP ident value doesn't match anything that has been recorded by NAT. Therefore NAT can't use the existing translation. It also can't create a new translation since it already has a fully extended translation entry and it doesn't have the ICMP ident to create another one. NAT drops packet 2 fragment 1.
4. Packet 2 fragment 0 arrives. NAT can use the above translation since the ICMP ident matches. (All pings within a single set of pings use the same ICMP ident number.) At this point, NAT records the IP ident of this packet. NAT translates and forwards packet 2 fragment 0.
5. Packet 2 fragment 2 arrives. NAT can now use the above translation since its IP ident value matches the one NAT recorded in the previous step. NAT translates and forwards packet 2 fragment 2. The destination device receives only fragment 0 and 2 (fragment 1 is missing), so it sends no reply.

Case 2

In this scenario, we see that if fragments other than the first fragment (fragment 0) arrive first, NAT creates a simple translation as long as there is an address in the NAT pool that has not already been used in a fully extended translation.

As we start, there is only one address in the NAT pool, the NAT translation table is empty, and the configuration appears as:

```
ip nat pool POOL1 10.10.10.3 10.10.10.3 prefix-length 24
ip nat inside source list 5 pool POOL1 overload
access-list 5 permit 172.16.0.0 0.0.0.31
```

1. Packet 1 fragment 1 arrives. NAT can't create a fully extended translation in the translation table since it doesn't have the ICMP ident information in this fragment. However, since there aren't any fully extended translations in place, NAT enters a simple translation. NAT then translates and forwards packet 1 fragment 1. The translation entry appears as:

Pro	Inside global	Inside local	Outside local	Outside global
---	10.10.10.3	172.16.0.1	---	---

2. Packet 1 fragment 0 arrives. Since the ICMP ident information is included in this fragment, NAT enters a fully extended translation entry:

Pro	Inside global	Inside local	Outside local	Outside global
---	10.10.10.3	172.16.0.1	---	---
icmp	10.10.10.3:24321	172.16.0.1:24321	172.17.1.2:24321	172.17.1.2:24321

NAT then records the IP ident information, and translates and forwards packet 1 fragment 0.

3. Packet 1 fragment 2 arrives. Because this fragment has the same IP ident information that NAT recorded in step 2, NAT uses the fully extended translation to translate and forward packet 1 fragment 2.

The destination device receives all fragments and replies. At this point, all pings succeed until the NAT translation table is cleared or times out.

Case 3

In this scenario, we see that if fragments other than the first fragment (fragment 0) arrive first, NAT creates a simple translation as long as there is an address in the NAT pool that has not already been used in a fully extended translation. If an extended translation in the NAT table already uses the address, you run the risk of NAT translating each of the fragment source addresses to a different address.

As we start, more than one address in the NAT pool performs overload, the translation table already has an extended translation in place, and the configuration is:

```
ip nat pool POOL1 10.10.10.3 10.10.10.5 prefix-length 24
ip nat inside source list 5 pool POOL1 overload
access-list 5 permit 172.16.0.0 0.0.0.31
```

The translation table appears as:

Pro	Inside global	Inside local	Outside local	Outside global
icmp	10.10.10.3:24322	172.16.0.1:24322	172.17.1.2:24322	172.17.1.2:24322

1. Packet 1 fragment 1 arrives. NAT can not create a fully extended translation table entry since it doesn't have the ICMP ident information in this fragment, and it can't create a simple translation entry for address 10.10.10.3, since there's an existing extended entry for this IP address. NAT picks the next free IP address (10.10.10.4) and creates a simple translation. NAT then translates and forwards packet 1 fragment 1. The translation table now appears as:

Pro	Inside global	Inside local	Outside local	Outside global
---	10.10.10.4	172.16.0.1	---	---
icmp	10.10.10.3:24322	172.16.0.1:24322	172.17.1.2:24322	172.17.1.2:24322

2. Packet 1 fragment 0 arrives. Since the ICMP ident information is included in this fragment, NAT enters a fully extended translation entry for address 10.10.10.3, and records the IP ident information for this packet. NAT then translates and forwards packet 1 fragment 0. The translation table now appears as:

Pro	Inside global	Inside local	Outside local	Outside global
---	10.10.10.4	172.16.0.1	---	---
icmp	10.10.10.3:24322	172.16.0.1:24322	172.17.1.2:24322	172.17.1.2:24322
icmp	10.10.10.3:24323	172.16.0.1:24323	172.17.1.2:24323	172.17.1.2:24323

3. Packet 1 fragment 2 arrives. Since its IP ident information matches the one NAT recorded in step 2, NAT uses the fully extended translation created in step 2 to translate and forward packet 1 fragment 2.

At this point, the destination device receives all fragments of packet 1, but fragment 0 and 2 have had their source address translated to 10.10.10.3 and fragment 1 has been translated to 10.10.10.4. Therefore, the destination device can not reassemble the packet and sends no reply.

4. Packet 2 fragment 0 arrives. NAT either uses the above fully extended translation or creates a new fully extended translation depending on the value of the fragment ICMP ident field. In either case, NAT records the IP ident information. NAT then translates and forwards packet 2 fragment 0.
5. Packet 2 fragment 2 arrives. Its IP ident information matches what NAT recorded in step 4, so NAT uses the second fully extended translation created in step 4. NAT translates and forwards packet 2 fragment 2.
6. Packet 2 fragment 1 arrives. Its IP ident information matches what NAT recorded in step 4, so NAT uses the second fully extended translation created in step 4. NAT translates and forwards packet 2 fragment 1.

The destination device receives all three fragments of packet 2 from the same source (10.10.10.3), so it reassembles the packet and replies.

Summary

Whether NAT drops or forwards an ICMP fragment depends on a number of things, such as the order in which the NAT router receives the fragments, and the state of the translation table at that time. Under certain conditions, NAT translates the fragments differently, which makes it impossible for the destination device to reassemble the packet.

NetPro Discussion Forums – Featured Conversations

Networking Professionals Connection is a forum for networking professionals to share questions, suggestions, and information about networking solutions, products, and technologies. The featured links are some of the most recent conversations available in this technology.

NetPro Discussion Forums – Featured Conversations for RP
Service Providers: MPLS
Virtual Private Networks: Services
Virtual Private Networks: Security

Related Information

- [NAT Support Page](#)
- [IP Routing Support Page](#)
- [Technical Support – Cisco Systems](#)

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2008 – 2009 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Aug 10, 2005

Document ID: 13771