Abstract

The Port Control Protocol (PCP) allows clients to request mappings in NAT gateways and firewalls. This document specifies the PCP UNSUPP_FAMILY error code, which enables PCP servers to inform clients when the requested external address family is not supported. This document also removes the requirement for the PCP server to validate the mapping nonce, which proved to be unhelpful in practice.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 24, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of
1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" [RFC2119].

2. PCP Unsupported Family Error

Port Control Protocol [RFC6887] MAP requests allow clients to request inbound mappings in NAT gateways and firewalls. A client can request a MAP mapping to an external IPv6 address or to an external IPv4 address. The client signifies which family of external address it desires by the type of address it puts into the Suggested External Address field.

If the client wants an external IPv6 address, then it populates the Suggested External Address field with a native IPv6 address. In the overwhelmingly common case where the client doesn’t know the external address when it makes its initial request, this will be the all-zeros IPv6 address (::).

If the client wants an external IPv4 address, then it populates the Suggested External Address field with an IPv4-mapped IPv6 address (the first 80 bits set to zero, the next 16 set to one). In the overwhelmingly common case where the client doesn’t know the NAT’s external address when it makes its initial request, this will be the all-zeros IPv4 address (::ffff:0:0).

The PCP specification [RFC6887] is somewhat vague about whether the address family is a firm requirement, or merely a hint that the PCP server is free to ignore. This update clarifies that issue: The specific address placed in the Suggested External Address field is merely a suggestion that the PCP server is free to ignore, but the address family is not. If the specific suggested address cannot be provided, another address of the same family SHOULD be provided if possible, but if the suggested address "family" cannot be provided by this PCP server, it MUST return a PCP error reply containing the UNSUPP_FAMILY error code.

Many gateway devices, particularly early ones, may not be able to provide both external address families. For example, an IPv4-only NAT cannot provide an external IPv6 address.
Even with gateway devices that can support both external address families, the ability to provide an external address of the requested family may depend on the family of the client’s internal address. For example, a gateway that supports native IPv6, and traditional NAT44, but not NAT64, can provide mappings from an internal IPv6 address to an external IPv6 address (typically the same address when no address translation is being performed), and can provide mappings from an internal IPv4 address to an external IPv4 address, but not mappings from an internal IPv6 address to an external IPv4 address. When such a gateway receives a request to map an internal IPv6 address to an external IPv4 address it MUST return the UNSUPP_FAMILY error code.

Note that it is possible and valid for a given internal address and port to have two mappings simultaneously, one to an external IPv4 address and one to an external IPv6 address. The handling of outbound packets is determined by the outbound destination address; for example, an outbound IPv6 packet addressed to an IPv6 address in the NAT64 gateway’s IPv6 address pool is translated to the corresponding IPv4 packet before forwarding; an outbound IPv6 packet addressed to some other routable IPv6 address is forwarded unmodified.

A client that can handle both IPv6 and IPv4 external addresses MAY send two requests, and then determine its behavior based on the responses it receives. For example, if the client requests and receives an IPv6 external address, it might create a DNS AAAA record giving that IPv6 address. If the client requests and receives an IPv4 external address, it might create a DNS address record giving that IPv4 address. If the client requests and receives both families of external address, it might create both DNS records. Or, if one external address is sufficient for the client, then it MAY first request its preferred address family, and only if that fails with an UNSUPP_FAMILY error, request the other family.
2.1. Implications for RFC 6887

Various sections of the PCP specification [RFC6887] describe clients and servers identifying a MAP mapping by examining the three-tuple of { protocol, internal address, internal port } in a request or reply. For example:

If the internal port, protocol, and internal address match an existing static mapping (which will have no nonce), then a PCP reply is sent giving the external address and port of that static mapping, using the nonce from the PCP request. The server does not record the nonce.

It is possible that a mapping might already exist for a requested internal address, protocol, and port. If so, the PCP server takes the following actions...

If no mapping exists for the internal address, protocol, and port, and the PCP server is able to create a mapping using the suggested external address and port, it SHOULD do so.

After performing common PCP response processing, the response is further matched with a previously sent MAP request by comparing the internal IP address (the destination IP address of the PCP response, or other IP address specified via the THIRD_PARTY option), the protocol, the internal port, and the mapping nonce. Other fields are not compared, because the PCP server sets those fields.

Everywhere that the PCP specification [RFC6887] refers to using the "protocol, internal address, and internal port," to identify a particular inbound mapping, it should be read to mean the four-tuple of { protocol, internal address, internal port, external address family }.

PCP clients and servers that only support one external address family can continue to use the previous three-tuple { protocol, internal address, internal port } to identify inbound mappings, since they only support one external address family, and unilaterally reject MAP requests and responses containing the unsupported family. For PCP servers this means rejecting MAP requests containing the unsupported address family via the UNSUPP_FAMILY error code. For PCP clients this should be a non-issue because a PCP client should never receive a reply containing an external address family it didn’t request, but should a client receive such a reply from a misbehaving PCP server offering an external address family the client did not request, the client MUST silently ignore the erroneous reply.
An implication of this update to the PCP specification is that when renewing a MAP mapping, a PCP client MUST include a suggested external address of the correct family, so that the gateway device can identify which mapping is being renewed. Ideally a PCP client SHOULD record the previously-granted external address and use that as the suggested external address in its renewal request, to facilitate recovery in the event of gateway state loss, but at the very least a PCP client MUST provide an all-zeroes suggested external address of the correct family (just as it must have indicated the desired address family in its initial request that created the mapping).

These considerations apply only to MAP requests. With PEER requests, the five-tuple of (protocol, internal address, internal port, remote peer address, remote peer port) uniquely identifies the intended mapping. When technologies like NAT64 are used the external address family need not be the same as the remote peer address family, but the external address family is still uniquely determined by the remote peer address, and does not need to be specified separately.
3. New Nonce Check Behavior

The PCP specification [RFC6887] states that if a client requests a mapping (or renews a mapping, which is the same thing, from the server’s point of view) and the requested mapping already exists, but with a different nonce, then the server returns a NOT_AUTHORIZED error.

This has proved to be problematic. The nonce exists to guard against off-path attackers. It helps a client have confidence that the PCP responses it receives are really from the server that processed its PCP request. And it helps a PCP server validate that a client requesting a mapping is the same client that previously requested a mapping for that internal address and port. In some circumstances a legitimate client may not know the correct nonce to renew its own mappings.

For example, if a host reboots or otherwise suffers a loss of state, it may not have a record of nonces it previously used. Suppose this host then requests a mapping from an external IPv4 address to its internal IP address at TCP port 22, so that it can receive ssh logins. If the same internal host had previously requested such a mapping using a different nonce, then the new request will fail with a NOT_AUTHORIZED error. This is unhelpful and misleading. The client does in fact have a mapping. Incoming connection requests to its external address and port will in fact be forwarded to it at port 22. The PCP server is simply refusing to tell the client what the external address and port are, hindering the client’s ability to use the mapping that it actually already has.

The same scenario also exists in the case where (i) a different internal host had previously requested a mapping to its internal port 22, (ii) that host then left the network, and (iii) the newly vacated internal IP address is then assigned to new host. When this happens, the new host will be unable to usefully request a mapping to its internal port 22 until the old mapping expires, or is deleted through some other means (e.g. via the DHCP server informing the PCP server that the IP address has been reassigned, or via manual intervention by an administrator, or via some other out-of-band mechanism). Note that the new host will actually have a working mapping to its internal port 22, and will actually receive incoming connection requests arriving at the external address and port, but the PCP server will refuse to tell the client what the external address and port are, thereby hindering the new host from communicating that external address and port to the peer it wishes to receive connections from. This is not helpful.

This PCP security check does not prevent the new host from learning
the external address and port by other circuitous means. For example, the new host could discover the external address and port by sending outbound traffic to a destination it controls, and having that destination report back the source address and port.

Furthermore, this PCP security check is inconsistent with other PCP behavior. It makes PCP behave differently for explicit dynamic versus other kinds of mappings. Indeed, requests matching static mappings are not subjected to the nonce check and will result in a response containing the static mapping’s current state. There is no reason that MAP requests matching a dynamic mapping should return less information.

Therefore, the nonce check behavior described below MUST be implemented instead.

3.1. Nonce Check for MAP Requests

If operating in the Simple Threat Model (Section 18.1 of the PCP specification [RFC6887]), and the internal port, protocol, internal address, and external address family match an existing explicit dynamic mapping, but the mapping nonce does not match, then the existing mapping is not modified in any way, and a valid PCP reply is returned to the client, using the client-specified nonce, reporting the external address, port, and remaining lifetime of the existing mapping.

This specification makes no statement about mapping nonce with the Advanced Threat Model.

3.2. Nonce Check for PEER Requests

If operating in the Simple Threat Model (Section 18.1 of the PCP specification [RFC6887]), and the protocol, internal address, internal port, remote peer address, and remote peer port match a mapping that already exists, but the mapping nonce does not match (that is, a previous PEER request was processed), then the existing mapping is not modified in any way, and a valid PCP reply is returned to the client, using the client-specified nonce, reporting the external address, port, and remaining lifetime of the existing mapping.

This specification makes no statement about mapping nonce with the Advanced Threat Model.
3.3. Returning NOT_AUTHORIZED error

A NOT_AUTHORIZED error should still be returned, as described in Section 15.1 of the PCP specification [RFC6887], when a PCP client attempts to delete a static mapping (i.e., a mapping created outside of PCP itself) or an outbound (implicit or PEER-created) mapping.

3.4. Discussion

The behavior described above in Sections 3.1 and 3.2 is what is currently being considered by the working group. An implication of this behavior is that if a client forgets its previous nonce (through reboot or similar lost of state), then when it tries to recreate its previous mappings, it will learn about its existing mappings, but it will be unable to extend their lifetimes. This means that a mapping with a one-hour lifetime will be renewed after roughly half an hour, at which point its remaining lifetime will be about half an hour. It will then be renewed after roughly fifteen minutes, then seven minutes, then three minutes, and so on, increasingly rapidly, until the old mapping finally expires and is immediately replaced with a new one with a new nonce.

The lower limit on the retry interval of four seconds implies that after a mapping expires, there will be a window of up to four seconds where no mapping exists, before the legitimate client re-tries its request and recreates the intended mapping (this time with the new nonce).

As an alternative to returning the current port and lifetime information about the mapping, the PCP server could instead return a NOT_AUTHORIZED error. However, were the PCP server to do this, the user is likely to perceive the gateway as "broken" and power-cycle it to fix the problem. Such forced reboot would clear out NAT state, thereby allowing a subsequent request to succeed, thereby (apparently) solving the problem. A pattern of habitual rebooting of the gateway to make it work gives the impression that the software is buggy and unreliable, and does not result in a positive user experience.
4. IANA Considerations

IANA should allocate the following PCP Result Code:

14 UNSUPP_FAMILY: Unsupported external address family, e.g., IPv6 in a NAT that handles only IPv4. This is a long lifetime error.

5. Security Considerations

The UNSUPP_FAMILY error code leaks no sensitive information and creates no new security vulnerabilities.

Allowing a client to learn the parameters of an existing mapping without knowing the mapping nonce used to create it could leak mapping information to an on-path attacker.

Having the PCP server refuse to renew or delete mappings if the request nonce doesn’t match the existing nonce allows an off-path attacker to preemptively poison a NAT gateway with bogus mappings, which the legitimate holder of the internal address will then be unable to renew or delete because it doesn’t know the nonce the attacker used when creating the bogus mappings.

6. Normative References


Authors’ Addresses

Stuart Cheshire
Apple Inc.
1 Infinite Loop
Cupertino, California 95014
USA

Phone: +1 408 974 3207
Email: cheshire@apple.com