                    Port Control Protocol (PCP) Proxy Function

Abstract

   This document specifies a new Port Control Protocol (PCP) functional
   element: the PCP proxy. The PCP proxy relays PCP requests received
   from PCP clients to upstream PCP server(s). A typical deployment
   usage of this function is to help establish successful PCP
   communications for PCP clients that cannot be configured with the
   address of a PCP server located more than one hop away.

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Further information on
   Internet Standards is available in Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc7648.

Copyright Notice

Table of Contents

1. Introduction

   This document defines a new Port Control Protocol (PCP) [RFC6887]
   functional element: the PCP proxy. As shown in Figure 1, the
   PCP proxy is logically equivalent to a PCP client back-to-back with a
   PCP server. The "glue" between the two is what is specified in this
   document. Other than that "glue", the server and the client behave
   exactly like their regular counterparts.

   The PCP proxy is responsible for relaying PCP messages received from
   PCP clients to upstream PCP servers and vice versa.

   Whether or not the PCP proxy is co-located with a flow-aware function
   (e.g., NAT, firewall) is deployment specific.

```
                        ................
          +------+      : +------+------+ :    +------+
          |Client|-------:-|Server|Client|-:----|Server|
          +------+      : +------+------+ :    +------+
                        :      Proxy     :
                        ................
```
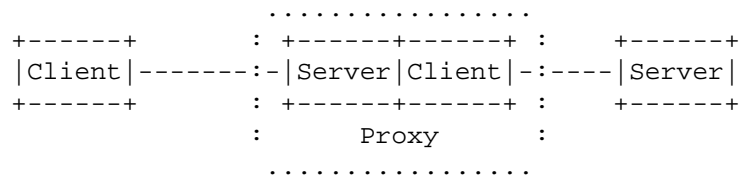
                   Figure 1: Reference Architecture

   This document assumes a hop-by-hop PCP authentication scheme. That
   is, referring to Figure 1, the leftmost PCP client authenticates with
   the PCP proxy, while the PCP proxy authenticates with the upstream
   server. Note that in some deployments, PCP authentication may only
   be enabled between the PCP proxy and an upstream PCP server (e.g., a
   customer premises host may not authenticate with the PCP proxy, but
   the PCP proxy may authenticate with the PCP server). The hop-by-hop
   authentication scheme is more suitable from a deployment standpoint.
   Furthermore, it allows implementations to easily support a PCP proxy
   that alters PCP messages (e.g., strips a PCP option, modifies a
   PCP field).

1.1. Use Case: The NAT Cascade

   In today's world, with public routable IPv4 addresses becoming less
   readily available, it is increasingly common for customers to receive
   a private address from their Internet Service Provider (ISP), and the
   ISP uses a NAT gateway of its own to translate those packets before
   sending them out onto the public Internet. This means that there is
   likely to be more than one NAT on the path between client machines
   and the public Internet:

   o If a residential customer receives a translated address from their
      ISP and then installs their own residential NAT gateway to share
      that address between multiple client devices in their home, then
      there are at least two NAT gateways on the path between client
      devices and the public Internet.

   o If a mobile phone customer receives a translated address from
      their mobile phone carrier and uses "Personal Hotspot" or
      "Internet Sharing" software on their mobile phone to make Wireless
      LAN (WLAN) Internet access available to other client devices, then
      there are at least two NAT gateways on the path between those
      client devices and the public Internet.

   o If a hotel guest connects a portable WLAN gateway to their hotel
      room's Ethernet port to share their room's Internet connection
      between their phone and their laptop computer, then packets from
      the client devices may traverse the hotel guest's portable NAT,
      the hotel network's NAT, and the ISP's NAT before reaching the
      public Internet.

   While it is possible, in theory, that client devices could somehow
   discover all the NATs on the path and communicate with each one
   separately using PCP [RFC6887], in practice it is not clear how
   client devices would reliably learn this information. Since the NAT
   gateways are installed and operated by different individuals and
   organizations, no single entity has knowledge of all the NATs on the
   path. Also, even if a client device could somehow know all the NATs
   on the path, requiring a client device to communicate separately with
   all of them imposes unreasonable complexity on PCP clients, many of
   which are expected to be simple low-cost devices.

   In addition, this goes against the spirit of NAT gateways. The main
   purpose of a NAT gateway is to make multiple downstream client
   devices appear, from the point of view of everything upstream of the
   NAT gateway, to be a single client device. In the same spirit, it
   makes sense for a PCP-capable NAT gateway to make multiple downstream

client devices requesting port mappings appear, from the point of
view of everything upstream of the NAT gateway, to be a single client
device requesting port mappings.

1.2. Use Case: The PCP Relay

Another envisioned use case of the PCP proxy is to help establish
successful PCP communications for PCP clients that cannot be
configured with the address of a PCP server located more than one hop
away. A PCP proxy can, for instance, be embedded in a CPE (Customer
Premises Equipment) while the PCP server is located in a network
operated by an ISP. This is illustrated in Figure 2.

```
                 |
     +------+    |
     |Client|--+
     +------+  |  +-----+                          +------+
              +--|Proxy|--------<ISP network>----------|Server|
     +------+  |  +-----+                          +------+
     |Client|--+     CPE
     +------+  |
               |
           LAN
```
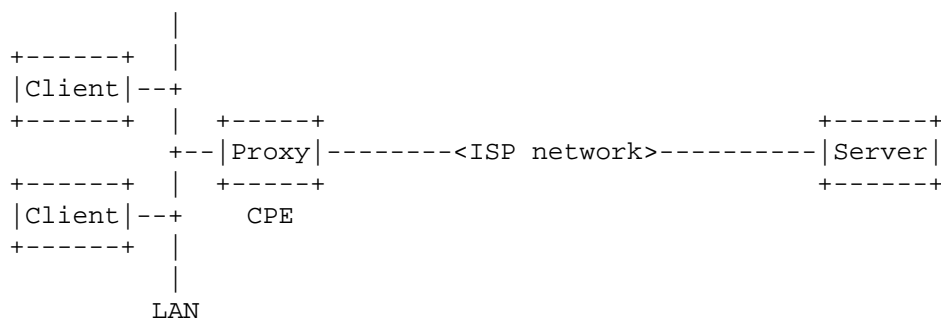
                  Figure 2: PCP Relay Use Case

This works because the proxy's server side is listening on the
address used as a default gateway by the clients. The clients use
that address as a fallback when discovering the PCP server's address.
The proxy picks up the requests and forwards them upstream to the
ISP's PCP server, with whose address it has been provisioned through
regular PCP client provisioning means.

This particular use case assumes that provisioning the server's
address on the CPE is feasible while doing it on the clients in the
LAN is not, which is what makes the PCP proxy valuable.

An alternative way to contact an upstream PCP server that may be
several hops away is to use a well-known anycast address
[PCP-ANYCAST], but that technique can be problematic when multiple
PCP servers are to be contacted [PCP-DEPLOY].

2. 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].

Where this document uses the terms "upstream" and "downstream", the
term "upstream" refers to the direction outbound packets travel
towards the public Internet, and the term "downstream" refers to the
direction inbound packets travel from the public Internet towards
client systems. Typically, when a home user views a web site, their
computer sends an outbound TCP SYN packet upstream towards the public
Internet, and an inbound downstream TCP SYN ACK reply comes back from
the public Internet.

3. Operation of the PCP Proxy

Upon receipt of a PCP mapping-creation request from a downstream
PCP client, a PCP proxy first examines its local mapping table to see
if it already has a valid active mapping matching the internal
address and internal port (and in the case of PEER requests, the
remote peer) given in the request.

If the PCP proxy does not already have a valid active mapping for
this mapping-creation request, then it allocates an available port on
its external interface. We assume for the sake of this description
that the address of its external interface is itself a private
address, subject to translation by an upstream NAT. The PCP proxy
then constructs an appropriate corresponding PCP request of its own
(as described below) and sends it to its upstream NAT, and the newly
created local mapping is considered temporary until a confirming
reply is received from the upstream PCP server.

If the PCP proxy does already have a valid active mapping for this
mapping-creation request and the lifetime remaining on the local
mapping is at least 3/4 of the lifetime requested by the PCP client,
then the PCP proxy SHOULD send an immediate reply giving the
outermost external address and port (previously learned using PCP
recursively, as described below) and the actual lifetime remaining
for this mapping. If the lifetime remaining on the local mapping is
less than 3/4 of the lifetime requested by the PCP client, then the
PCP proxy MUST generate an upstream request as described below.

For mapping-deletion requests (lifetime = 0), the local mapping, if
any, is deleted, and then (regardless of whether or not a local
mapping existed) a corresponding upstream request is generated.

The PCP proxy knows the destination IP address for its upstream
PCP request using the same means that are available for provisioning
a PCP client. In particular, the PCP proxy MUST follow the procedure
defined in Section 8.1 of the PCP specification [RFC6887] to discover
its PCP server. This does not preclude other means from being used
in addition.

   In the upstream PCP request:

   o The PCP client's IP address and internal port are the PCP proxy's
      own external address and port just allocated for this mapping.

   o The suggested external address and port in the upstream
      PCP request SHOULD be copied from the original PCP request. On a
      typical renewal request, this will be the outermost external
      address and port previously learned by the client.

   o The requested lifetime is as requested by the client if it falls
      within the acceptable range for this PCP server; otherwise, it
      SHOULD be capped to appropriate minimum and maximum values
      configured for this PCP server.

   o The mapping nonce is copied from the original PCP request.

   o For PEER requests, the remote peer IP address and port are copied
      from the original PCP request.

   Upon receipt of a PCP reply giving the outermost (i.e., publicly
   routable) external address, port, and lifetime, the PCP proxy records
   this information in its own mapping table and relays the information
   to the requesting downstream PCP client in a PCP reply. The
   PCP proxy therefore records, among other things, the following
   information in its mapping table:

   o Client's internal address and port.

   o External address and port allocated by this PCP proxy.

   o Outermost external address and port allocated by the upstream
      PCP server.

   o Mapping lifetime (also dictated by the upstream PCP server).

   o Mapping nonce.

   In the downstream PCP reply:

   o The lifetime is as granted by the upstream PCP server, or less if
      the granted lifetime exceeds the maximum lifetime this PCP server
      is configured to grant. If the proxy chooses to grant a
      downstream lifetime greater than the lifetime granted by the
      upstream PCP server (which is NOT RECOMMENDED), then this
      PCP proxy MUST take responsibility for renewing the upstream
      mapping itself.

   o The Epoch Time is this PCP proxy's Epoch Time, not the Epoch Time
      of the upstream PCP server. Each PCP server has its own
      independent Epoch Time. However, if the Epoch Time received from
      the upstream PCP server indicates a loss of state in that
      PCP server, the PCP proxy can either (1) recreate the lost
      mappings itself or (2) reset its own Epoch Time to cause its
      downstream clients to perform such state repairs themselves. A
      PCP proxy MUST NOT simply copy the upstream PCP server's
      Epoch Time into its downstream PCP replies, because if it suffers
      its own state loss it needs the ability to communicate that state
      loss to clients. Thus, each PCP server has its own independent
      Epoch Time. However, as a convenience, a downstream PCP proxy may
      simply choose to reset its own Epoch Time whenever it detects that
      its upstream PCP server has lost state. Thus, in this case, the
      PCP proxy's Epoch Time always resets whenever its upstream
      PCP server loses state; it may reset at other times as well.

   o The mapping nonce is copied from the reply received from the
      upstream PCP server.

   o The assigned external port and assigned external IP address are
      copied from the reply received from the upstream PCP server (i.e.,
      they are the outermost external IP address and port, not the
      locally assigned external address and port). By recursive
      application of this procedure, the outermost external IP address
      and port are relayed from the outermost NAT, through one or more
      intervening PCP proxies, until they ultimately reach the
      downstream client.

   o For PEER requests, the remote peer IP address and port are copied
      from the reply received from the upstream PCP server.

3.1. Optimized Hairpin Routing

   A PCP proxy SHOULD implement optimized hairpin routing. What this
   means is the following:

   o If a PCP proxy observes an outgoing packet arriving on its
      internal interface that is addressed to an external address and
      port appearing in the NAT gateway's own mapping table, then the
      NAT gateway SHOULD (after creating a new outbound mapping if one
      does not already exist) rewrite the packet appropriately and
      deliver it to the internal client to which that external address
      and port are currently allocated.

o Similarly, if a PCP proxy observes an outgoing packet arriving on
   its internal interface that is addressed to an *outermost*
   external address and port appearing in the NAT gateway's own
   mapping table, then the NAT gateway SHOULD do as described above:
   create a new outbound mapping if one does not already exist, and
   then rewrite the packet appropriately and deliver it to the
   internal client to which that outermost external address and port
   are currently allocated. This is not necessary for successful
   communication, but it provides efficiency. Without this optimized
   hairpin routing, the packet will be delivered all the way to the
   outermost NAT gateway, which will then perform standard hairpin
   translation and send it back. Using knowledge of the outermost
   external address and port, this rewriting can be anticipated and
   performed locally. This rewriting technique will typically offer
   higher throughput and lower latency than sending packets all the
   way to the outermost NAT gateway and back.

Note that traffic counters maintained by an upstream PCP server will
differ from the counters of a PCP proxy implementing optimized
hairpin routing.

3.2. Termination of Recursion

Any recursive algorithm needs a mechanism to terminate the recursion
at the appropriate point. This termination of recursion can be
achieved in a variety of ways. The following (non-exhaustive)
examples are provided for illustration purposes:

o An ISP's PCP-controlled gateway (which may embed a NAT, firewall,
   or any function that can be controlled with PCP) could be
   configured to know that it is the outermost PCP-controlled
   gateway, and consequently it does not need to relay PCP requests
   upstream.

o A PCP-controlled gateway could determine automatically that if its
   external address is not one of the known private addresses
   [RFC1918] [RFC6598], then its external address is a public
   routable IP address, and consequently it does not need to relay
   PCP requests upstream.

o Recursion may be terminated if there is no explicit list of
   PCP servers configured (manually, using DHCP [RFC7291], or
   otherwise) or if its default router is not responsive to
   PCP requests.

o Recursion may also be terminated if the upstream PCP-controlled
   device does not embed a PCP proxy.

3.3. Source Address for PCP Requests Sent Upstream

   As with a regular PCP server, the PCP-controlled device can be a NAT,
   a firewall, or even some sort of hybrid. In particular, a PCP proxy
   that simply relays all requests upstream can be thought of as the
   degenerate case of a PCP server controlling a wide-open firewall
   back-to-back with a regular PCP client.

   One important property of the PCP-controlled device will affect the
   PCP proxy's behavior: when the proxy's server part instructs the
   device to create a mapping, that mapping's external address may or
   may not be one that belongs to the proxy node.

   o When the mapping's external address belongs to the proxy node, as
      would presumably be the case for a NAT, then the proxy's client
      side sends out an upstream PCP request using the mapping's
      external IP address as the source.

   o When the mapping's external address does not belong to the proxy
      node, as would presumably be the case for a firewall, then the
      proxy's client side needs to install upstream mappings on behalf
      of its downstream clients. To do this, it MUST insert a
      THIRD_PARTY option in its upstream PCP request carrying the
      mapping's external address.

   Note that hybrid PCP-controlled devices may create NAT-like mappings
   in some circumstances and firewall-like mappings in others. A proxy
   controlling such a device would adjust its behavior dynamically,
   depending on the kind of mapping created.

3.4. Unknown Opcodes and Options

3.4.1. No NAT Is Co-located with the PCP Proxy

   When no NAT is co-located with the PCP proxy, the port numbers
   included in received PCP messages (from the PCP server or
   PCP client(s)) are not altered by the PCP proxy. The PCP proxy
   relays to the PCP server unknown options and Opcodes because there is
   no reachability failure risk.

3.4.2. PCP Proxy Co-located with a NAT Function

   By default, the proxy MUST relay unknown Opcodes and mandatory-to-
   process unknown options. Rejecting unknown options and Opcodes has
   the drawback of preventing a PCP client from making use of new
   capabilities offered by the PCP server but not supported by the
   PCP proxy, even if no IP address and/or port is included in the
   option/Opcode.

   Because PCP messages with an unknown Opcode or mandatory-to-process
   unknown options can carry a hidden internal address or internal port
   that will not be translated, a PCP proxy MUST be configurable to
   disable relaying unknown Opcodes and mandatory-to-process unknown
   options. If the PCP proxy is configured to disable relaying unknown
   Opcodes and mandatory-to-process unknown options, the PCP proxy MUST
   behave as follows:

   o a PCP proxy co-located with a NAT MUST reject, via an
      UNSUPP_OPCODE error response, a received request with an unknown
      Opcode.

   o a PCP proxy co-located with a NAT MUST reject, via an
      UNSUPP_OPTION error response, a received request with a mandatory-
      to-process unknown option.

3.5. Mapping Repair

   ANNOUNCE requests received from PCP clients are handled locally; as
   such, these requests MUST NOT be relayed to the provisioned
   PCP server.

   Upon receipt of an unsolicited ANNOUNCE response from a PCP server,
   the PCP proxy proceeds to renew the mappings and checks to see
   whether or not there are changes compared to a local cache if it is
   maintained by the PCP proxy. If no change is detected, no
   unsolicited ANNOUNCE is generated towards PCP clients. If a change
   is detected, the PCP proxy MUST generate unsolicited ANNOUNCE
   message(s) to appropriate PCP clients. If the PCP proxy does not
   maintain a local cache for the mappings, unsolicited multicast
   ANNOUNCE messages are sent to PCP clients.

   Upon change of its external IP address, the PCP proxy SHOULD renew
   the mappings it maintained. If the PCP server assigns a different
   external port, the PCP proxy SHOULD follow the PCP mapping repair
   procedure [RFC6887]. This can be achieved only if a full state table
   is maintained by the PCP proxy.

3.6. Multiple PCP Servers

   A PCP proxy MAY handle multiple PCP servers at the same time. Each
   PCP server is associated with its own epoch value. PCP clients are
   not aware of the presence of multiple PCP servers.

   Following the PCP Server Selection process [RFC7488], if several
   PCP servers are configured to the PCP proxy, it will contact in
   parallel all these PCP servers.

   In some contexts (e.g., PCP-controlled Carrier-Grade NATs (CGNs)),
   the PCP proxy MAY load-balance the PCP clients among available
   PCP servers. The PCP proxy MUST ensure that requests of a given
   PCP client are relayed to the same PCP server.

   The PCP proxy MAY rely on some fields (e.g., Zone-ID [PCP-ZONES]) in
   the PCP request to redirect the request to a given PCP server.

4. Security Considerations

   The PCP proxy MUST follow the security considerations detailed in the
   PCP specification [RFC6887] for both the client and server side.

   Section 3.3 specifies the cases where a THIRD_PARTY option is
   inserted by the PCP proxy. In those cases, ways to prevent a
   malicious user from creating mappings on behalf of a third party must
   be employed as discussed in Section 13.1 of the PCP specification
   [RFC6887]. In particular, THIRD_PARTY options MUST NOT be enabled
   unless the network on which the PCP messages are to be sent is fully
   trusted (via physical or cryptographic security, or both) -- for
   example, if access control lists (ACLs) are installed on the
   PCP proxy, the PCP server, and the network between them so that those
   ACLs allow only communications from a trusted PCP proxy to the
   PCP server.

   A received request carrying an unknown Opcode or option SHOULD be
   dropped (or, in the case of an unknown option that is not mandatory
   to process, the option SHOULD be removed) if it is not compatible
   with security controls provisioned to the PCP proxy.

   The device embedding the PCP proxy MAY block PCP requests directly
   sent to the upstream PCP server(s). This can be enforced using ACLs.

5. References

5.1. Normative References

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119,
              DOI 10.17487/RFC2119, March 1997,
              <http://www.rfc-editor.org/info/rfc2119>.

   [RFC6887] Wing, D., Ed., Cheshire, S., Boucadair, M., Penno, R., and
              P. Selkirk, "Port Control Protocol (PCP)", RFC 6887,
              DOI 10.17487/RFC6887, April 2013,
              <http://www.rfc-editor.org/info/rfc6887>.

5.2. Informative References

   [PCP-ANYCAST]
              Kiesel, S., Penno, R., and S. Cheshire, "Port Control
              Protocol (PCP) Anycast Addresses", Work in Progress,
              draft-ietf-pcp-anycast-07, August 2015.

   [PCP-DEPLOY]
              Boucadair, M., "Port Control Protocol (PCP) Deployment
              Models", Work in Progress,
              draft-boucadair-pcp-deployment-cases-03, July 2014.

   [PCP-ZONES]
              Penno, R., "PCP Support for Multi-Zone Environments", Work
              in Progress, draft-penno-pcp-zones-01, October 2011.

   [RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G.,
              and E. Lear, "Address Allocation for Private Internets",
              BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996,
              <http://www.rfc-editor.org/info/rfc1918>.

   [RFC6598] Weil, J., Kuarsingh, V., Donley, C., Liljenstolpe, C., and
              M. Azinger, "IANA-Reserved IPv4 Prefix for Shared Address
              Space", BCP 153, RFC 6598, DOI 10.17487/RFC6598, April
              2012, <http://www.rfc-editor.org/info/rfc6598>.

   [RFC7291] Boucadair, M., Penno, R., and D. Wing, "DHCP Options for
              the Port Control Protocol (PCP)", RFC 7291,
              DOI 10.17487/RFC7291, July 2014,
              <http://www.rfc-editor.org/info/rfc7291>.

   [RFC7488] Boucadair, M., Penno, R., Wing, D., Patil, P., and T.
              Reddy, "Port Control Protocol (PCP) Server Selection",
              RFC 7488, DOI 10.17487/RFC7488, March 2015,
              <http://www.rfc-editor.org/info/rfc7488>.

Authors' Addresses

   Simon Perreault
   Jive Communications
   Quebec, QC
   Canada

   Email: sperreault@jive.com

   Mohamed Boucadair
   France Telecom
   Rennes 35000
   France

   Email: mohamed.boucadair@orange.com

   Reinaldo Penno
   Cisco
   United States

   Email: repenno@cisco.com

   Dan Wing
   Cisco Systems, Inc.
   170 West Tasman Drive
   San Jose, California 95134
   United States

   Email: dwing@cisco.com

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

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