

PCP working group
Internet-Draft
Intended status: Standards Track
Expires: April 8, 2016

S. Kiesel
University of Stuttgart
R. Penno
Cisco Systems, Inc.
S. Cheshire
Apple
October 6, 2015

Port Control Protocol (PCP) Anycast Addresses
draft-ietf-pcp-anycast-08

Abstract

The Port Control Protocol (PCP) Anycast Addresses enable PCP clients to transmit signaling messages to their closest PCP-aware on-path NAT, Firewall, or other middlebox, without having to learn the IP address of that middlebox via some external channel. This document establishes one well-known IPv4 address and one well-known IPv6 address to be used as PCP Anycast Addresses.

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 8, 2016.

Copyright Notice

Copyright (c) 2015 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 the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. PCP Server Discovery based on well-known IP Address	4
2.1. PCP Discovery Client behavior	4
2.2. PCP Discovery Server behavior	4
3. Deployment Considerations	5
4. IANA Considerations	6
4.1. Registration of IPv4 Special Purpose Address	6
4.2. Registration of IPv6 Special Purpose Address	6
5. Security Considerations	7
5.1. Information Leakage through Anycast	7
5.2. Hijacking of PCP Messages sent to Anycast Addresses	7
6. Acknowledgments	9
7. References	10
7.1. Normative References	10
7.2. Informative References	10
Authors' Addresses	11

1. Introduction

The Port Control Protocol (PCP) [RFC6887] provides a mechanism to control how incoming packets are forwarded by upstream devices such as Network Address Translator IPv6/IPv4 (NAT64), Network Address Translator IPv4/IPv4 (NAT44), and IPv6 and IPv4 firewall devices. Furthermore, it provides a mechanism to reduce application keep alive traffic [I-D.ietf-pcp-optimize-keepalives]. The PCP base protocol document [RFC6887] specifies the message formats used, but the address to which a client sends its request is either assumed to be the default router (which is appropriate in a typical single-link residential network) or has to be configured otherwise via some external mechanism, such as a configuration file or a DHCP option [RFC7291].

This document follows a different approach: it establishes two well-known anycast addresses for the PCP Server, one IPv4 address and one IPv6 address. PCP clients usually send PCP requests to these well-known addresses if no other PCP server addresses are known or after communication attempts to such other addresses have failed. The anycast addresses are allocated from pools of special-purpose IP addresses (see Section 4), in accordance with Section 3.4 of [RFC4085]. Yet, a means to disable or override these well-known addresses (e. g., a configuration file option) should be available in implementations.

Using an anycast address is particularly useful in larger network topologies. For example, if the PCP-enabled NAT/firewall function is not located on the client's default gateway, but further upstream in a Carrier-grade NAT (CGN), sending PCP requests to the default gateway's IP address will not have the desired effect. When using a configuration file or the DHCP option to learn the PCP server's IP address, this file or the DHCP server configuration must reflect the network topology, and the router and CGN configuration. This may be cumbersome to achieve and maintain. If there is more than one upstream CGN and traffic is routed using a dynamic routing protocol such as OSPF, this approach may not be feasible at all, as it cannot provide timely information on which CGN to interact with. In contrast, when using the PCP anycast address, the PCP request will travel through the network like any other packet, without any special support from DNS, DHCP, other routers, or anything else, until it reaches the PCP-capable device, which receives it, handles it, and sends back a reply. A further advantage of using an anycast address instead of a DHCP option is, that the anycast address can be hard-coded into the application. There is no need for an application programming interface for passing the PCP server's address from the operating system's DHCP client to the application. For further discussion of deployment considerations see Section 3.

2. PCP Server Discovery based on well-known IP Address

2.1. PCP Discovery Client behavior

PCP clients can add the PCP anycast addresses, which are defined in Sections 4.1 and 4.2, after the default router list (for IPv4 and IPv6) to the list of PCP server(s) (see Section 8.1, step 2. of [RFC6887]). This list is processed as specified in [RFC7488].

Note: If, in some specific scenario, it was desirable to use only the anycast address (and not the default router), this could be achieved by putting the anycast address into the configuration file, or DHCP option, etc.

2.2. PCP Discovery Server behavior

PCP Servers can be configured to listen on the anycast addresses for incoming PCP requests. When a PCP server receives a PCP requests destined for an anycast address it supports, it sends the corresponding PCP replies using that same anycast address as the source address (see Page 6 of [RFC1546] for further discussion).

3. Deployment Considerations

For general recommendations regarding operation of anycast services see [RFC4786]. Architectural considerations of IP anycast are discussed in [RFC7094].

In some deployment scenarios, using PCP anycasting may have certain limitations, which can be overcome by using additional mechanisms or by using other PCP server discovery methods instead, such as DHCP [RFC7291] or a configuration file.

One important example is a network topology, in which a network is connected to one or more upstream network(s) via several parallel firewalls, each individually controlled by its own PCP server. Even if all of these PCP servers are configured for anycasting, only one will receive the messages sent by a given client, depending on the state of the routing tables.

As long as routing is always symmetric, i.e., all upstream and downstream packets from/to that client are routed through this very same firewall, communication will be possible as expected. If there is a routing change, a PCP client using PCP anycasting might start interacting with a different PCP server. From the PCP client's point of view this would be the same as a PCP server reboot and the client could detect it by examining the Epoch field during the next PCP response or ANNOUNCE message. The client would re-establish the firewall rules and packet flows could resume.

If, however, routing is asymmetric, upstream packets from a client traverse a different firewall than the downstream packets to that client. Establishing policy rules in only one of these two firewalls by means of PCP anycasting will not have the desired result of allowing bi-directional connectivity. One solution approach to overcome this problem is an implementation-specific mechanism to synchronize state between all firewalls at the border of a network, i.e., a PEER message sent to any of these PCP servers would establish rules in all firewalls. Another approach would be to use a different discovery mechanism (e.g., DHCP or a configuration file) that allows a PCP client to acquire a list of all PCP servers controlling the parallel firewalls and configure each of them individually.

4. IANA Considerations

4.1. Registration of IPv4 Special Purpose Address

IANA is requested to assign a single IPv4 address from the 192.0.0.0/24 prefix and register it in the IANA IPv4 Special-Purpose Address Registry [RFC6890].

Attribute	Value
Address Block	192.0.0.???/32 (??? = TBD by IANA)
Name	Port Control Protocol Anycast
RFC	This document, if approved (TBD)
Allocation Date	Date of approval of this document (TBD)
Termination Date	N/A
Source	True
Destination	True
Forwardable	True
Global	True
Reserved-by-Protocol	False

4.2. Registration of IPv6 Special Purpose Address

IANA is requested to assign a single IPv6 address from the 2001:0000::/23 prefix and register it in the IANA IPv6 Special-Purpose Address Registry [RFC6890].

Attribute	Value
Address Block	2001:0????????/128 (??? = TBD by IANA)
Name	Port Control Protocol Anycast
RFC	This document, if approved (TBD)
Allocation Date	Date of approval of this document (TBD)
Termination Date	N/A
Source	True
Destination	True
Forwardable	True
Global	True
Reserved-by-Protocol	False

5. Security Considerations

In addition to the security considerations in [RFC6887], [RFC4786], and [RFC7094], two further security issues are considered here.

5.1. Information Leakage through Anycast

In a network without any border gateway, NAT or firewall that is aware of the PCP anycast address, outgoing PCP requests could leak out onto the external Internet, possibly revealing information about internal devices.

Using an IANA-assigned well-known PCP anycast address enables border gateways to block such outgoing packets. In the default-free zone, routers should be configured to drop such packets. Such configuration can occur naturally via BGP messages advertising that no route exists to said address.

Sensitive clients that do not wish to leak information about their presence can set an IP TTL on their PCP requests that limits how far they can travel towards the public Internet. However, methods for choosing an appropriate TTL value, e.g., based on the assumed radius of the trusted network domain, is beyond the scope of this document.

Before sending PCP requests with possibly privacy-sensitive parameters (e.g., IP addresses and port numbers) to the PCP anycast addresses, PCP clients can send an ANNOUNCE request (without parameters; see Section 14.1 of [RFC6887]), in order to probe whether a PCP server consumes and processes PCP requests sent to that anycast address.

5.2. Hijacking of PCP Messages sent to Anycast Addresses

The anycast addresses are treated by normal host operating systems just as normal unicast addresses, i.e., packets destined for an anycast address are sent to the default router for processing and forwarding. Hijacking such packets in the first network segment would effectively require the attacker to impersonate the default router, e.g., by means of ARP spoofing in an Ethernet network. Once an anycast message is forwarded closer to the core network, routing will likely become subject to dynamic routing protocols such as OSPF or BGP. Anycast messages could be hijacked by announcing counterfeited messages in these routing protocols. When analyzing the risk and possible consequences of such attacks in a given network scenario, the probable impacts on PCP signaling need to be put into proportion with probable impacts on other protocols such as the actual application protocols.

In addition to following best current practices in first hop security and routing protocol security, PCP authentication [RFC7652] may be useful in some scenarios. However, the effort needed for a proper setup of this authentication mechanism (e.g., installing the right shared secrets or cryptographic keys on all involved systems) may thwart the goal of fully automatic configuration by using PCP anycast. Therefore, this approach may be less suitable for scenarios with high trust between the operator of the PCP-controlled middlebox and all users (e.g., a residential gateway used only by family members) or if there is anyway rather limited trust that the middlebox will behave correctly (e.g., the Wifi in an airport lounge). In contrast, this scheme may be highly useful in scenarios with many users and a trusted network operator, such as a large corporate network or a university campus network, which uses several parallel NATs or firewalls to connect to the Internet. Therefore, a thorough analysis of the benefits and costs of using PCP authentication in a given network scenario is recommended.

6. Acknowledgments

The authors would like to thank the members of the PCP working group for contributions and feedback, in particular Mohamed Boucadair, Charles Eckel, Simon Perreault, Tirumaleswar Reddy, Markus Stenberg, Dave Thaler, and Dan Wing.

7. References

7.1. Normative References

- [RFC6887] Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, April 2013.
- [RFC6890] Cotton, M., Vegoda, L., Bonica, R., and B. Haberman, "Special-Purpose IP Address Registries", BCP 153, RFC 6890, April 2013.
- [RFC7488] Boucadair, M., Penno, R., Wing, D., Patil, P., and T. Reddy, "Port Control Protocol (PCP) Server Selection", RFC 7488, March 2015.

7.2. Informative References

- [I-D.ietf-pcp-optimize-keepalives] Reddy, T., Patil, P., Isomaki, M., and D. Wing, "Optimizing NAT and Firewall Keepalives Using Port Control Protocol (PCP)", draft-ietf-pcp-optimize-keepalives-06 (work in progress), May 2015.
- [RFC1546] Partridge, C., Mendez, T., and W. Milliken, "Host Anycasting Service", RFC 1546, November 1993.
- [RFC4085] Plonka, D., "Embedding Globally-Routable Internet Addresses Considered Harmful", BCP 105, RFC 4085, DOI 10.17487/RFC4085, June 2005, <<http://www.rfc-editor.org/info/rfc4085>>.
- [RFC4786] Abley, J. and K. Lindqvist, "Operation of Anycast Services", BCP 126, RFC 4786, December 2006.
- [RFC7094] McPherson, D., Oran, D., Thaler, D., and E. Osterweil, "Architectural Considerations of IP Anycast", RFC 7094, DOI 10.17487/RFC7094, January 2014, <<http://www.rfc-editor.org/info/rfc7094>>.
- [RFC7291] Boucadair, M., Penno, R., and D. Wing, "DHCP Options for the Port Control Protocol (PCP)", RFC 7291, July 2014.
- [RFC7652] Cullen, M., Hartman, S., Zhang, D., and T. Reddy, "Port Control Protocol (PCP) Authentication Mechanism", RFC 7652, DOI 10.17487/RFC7652, September 2015, <<http://www.rfc-editor.org/info/rfc7652>>.

Authors' Addresses

Sebastian Kiesel
University of Stuttgart Information Center
Networks and Communication Systems Department
Allmandring 30
Stuttgart 70550
Germany

Email: ietf-pcp@skiesel.de

Reinaldo Penno
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134
USA

Email: repenno@cisco.com

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

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

Network Working Group
Internet-Draft
Updates: 6887 (if approved)
Intended status: Standards Track
Expires: January 21, 2016

M. Wasserman
S. Hartman
Painless Security
D. Zhang
Huawei
T. Reddy
Cisco
July 20, 2015

Port Control Protocol (PCP) Authentication Mechanism
draft-ietf-pcp-authentication-14

Abstract

An IPv4 or IPv6 host can use the Port Control Protocol (PCP) to flexibly manage the IP address and port mapping information on Network Address Translators (NATs) or firewalls to facilitate communication with remote hosts. However, the un-controlled generation or deletion of IP address mappings on such network devices may cause security risks and should be avoided. In some cases the client may need to prove that it is authorized to modify, create or delete PCP mappings. This document describes an in-band authentication mechanism for PCP that can be used in those cases. The Extensible Authentication Protocol (EAP) is used to perform authentication between PCP devices.

This document updates RFC6887.

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 January 21, 2016.

Copyright Notice

Copyright (c) 2015 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 the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
2.	Terminology	4
3.	Protocol Details	5
3.1.	Session Initiation	5
3.1.1.	Authentication triggered by the client	6
3.1.2.	Authentication triggered by the server	7
3.1.3.	Authentication using EAP	7
3.2.	Recovery from lost PA session	9
3.3.	Session Termination	10
3.4.	Session Re-Authentication	11
4.	PA Security Association	12
5.	Packet Format	13
5.1.	Packet Format of PCP Auth Messages	13
5.2.	Opcode-specific information of AUTHENTICATION Opcode	15
5.3.	NONCE Option	16
5.4.	AUTHENTICATION_TAG Option	16
5.5.	PA_AUTHENTICATION_TAG option	18
5.6.	EAP_PAYLOAD Option	19
5.7.	PRF Option	19
5.8.	MAC_ALGORITHM Option	20
5.9.	SESSION_LIFETIME Option	20
5.10.	RECEIVED_PAK Option	21
5.11.	ID_INDICATOR Option	21
6.	Processing Rules	22
6.1.	Authentication Data Generation	22
6.2.	Authentication Data Validation	23
6.3.	Retransmission Policies for PA Messages	24
6.4.	Sequence Numbers for PCP Auth Messages	24
6.5.	Sequence Numbers for Common PCP Messages	25
6.6.	MTU Considerations	26
7.	IANA Considerations	27

7.1.	NONCE	28
7.2.	AUTHENTICATION_TAG	28
7.3.	PA_AUTHENTICATION_TAG	28
7.4.	EAP_PAYLOAD	29
7.5.	PRF	29
7.6.	MAC_ALGORITHM	29
7.7.	SESSION_LIFETIME	30
7.8.	RECEIVED_PAK	30
7.9.	ID_INDICATOR	30
8.	Security Considerations	31
9.	Acknowledgements	31
10.	Change Log	32
10.1.	Changes from wasserman-pcp-authentication-02 to ietf-pcp-authentication-00	32
10.2.	Changes from wasserman-pcp-authentication-01 to -02	32
10.3.	Changes from ietf-pcp-authentication-00 to -01	32
10.4.	Changes from ietf-pcp-authentication-01 to -02	32
10.5.	Changes from ietf-pcp-authentication-02 to -03	33
10.6.	Changes from ietf-pcp-authentication-03 to -04	33
10.7.	Changes from ietf-pcp-authentication-04 to -05	33
10.8.	Changes from ietf-pcp-authentication-05 to -06	33
11.	References	34
11.1.	Normative References	34
11.2.	Informative References	35
	Authors' Addresses	35

1. Introduction

Using the Port Control Protocol (PCP) [RFC6887], an application can flexibly manage the IP address mapping information on its network address translators (NATs) and firewalls, and control their policies in processing incoming and outgoing IP packets. Because NATs and firewalls both play important roles in network security architectures, there are many situations in which authentication and access control are required to prevent un-authorized users from accessing such devices. This document defines a PCP security extension that enables PCP servers to authenticate their clients with Extensible Authentication Protocol (EAP). The EAP messages are encapsulated within PCP messages during transportation.

The following issues are considered in the design of this extension:

- o Loss of EAP messages during transportation
- o Reordered delivery of EAP messages
- o Generation of transport keys

- o Integrity protection and data origin authentication for PCP messages
- o Algorithm agility

The mechanism described in this document meets the security requirements to address the Advanced Threat Model described in the base PCP specification [RFC6887]. This mechanism can be used to secure PCP in the following situations:

- o On security infrastructure equipment, such as corporate firewalls, that do not create implicit mappings for specific traffic.
- o On equipment (such as CGNs or service provider firewalls) that serve multiple administrative domains and do not have a mechanism to securely partition traffic from those domains.
- o For any implementation that wants to be more permissive in authorizing applications to create mappings for successful inbound communications destined to machines located behind a NAT or a firewall.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Most of the terms used in this document are introduced in [RFC6887].

PCP Client: A PCP software instance that is responsible for issuing PCP requests to a PCP server. In this document, a PCP client is also a EAP peer [RFC3748], and it is the responsibility of a PCP client to provide the credentials when authentication is required.

PCP Server: A PCP software instance that resides on the PCP-Controlled Device that receives PCP requests from the PCP client and creates appropriate state in response to that request. In this document, a PCP server is integrated with an EAP authenticator [RFC3748]. Therefore, when necessary, a PCP server can verify the credentials provided by a PCP client and make an access control decision based on the authentication result.

PCP-Authentication (PA) Session: A series of PCP message exchanges transferred between a PCP client and a PCP server. The PCP messages involved within a session includes the PA messages used to perform EAP authentication, key distribution and session management, and the common PCP messages secured with the keys distributed during

authentication. Each PA session is assigned a distinctive Session ID.

Session Partner: A PCP implementation involved within a PA session. Each PA session has two session partners (a PCP server and a PCP client).

PCP device: A PCP client or a PCP server.

Session Lifetime: The lifetime associated with a PA session, which decides the lifetime of the current authorization given to the PCP client.

PCP Security Association (PCP SA): A PCP security association is formed between a PCP client and a PCP server by sharing cryptographic keying material and associated context. The formed duplex security association is used to protect the bidirectional PCP signaling traffic between the PCP client and PCP server.

Master Session Key (MSK): A key derived by the partners of a PA session, using an EAP key generating method (e.g., the one defined in [RFC5448]).

PCP-Authentication (PA) message: A PCP message containing an AUTHENTICATION Opcode. Particularly, a PA message sent from a PCP server to a PCP client is referred to as a PA-Server message, while a PA message sent from a PCP client to a PCP server is referred to as a PA-Client message. Therefore, a PA-Server message is actually a PCP response message specified in [RFC6887], and a PA-Client message is a PCP request message. This document specifies an option, the PA_AUTHENTICATION_TAG Option defined in Section 5.5 for PCP authentication, to provide integrity protection and message origin authentication for PA messages.

Common PCP message: A PCP message which does not contain an AUTHENTICATION Opcode. This document specifies an AUTHENTICATION_TAG Option to provide integrity protection and message origin authentication for the common PCP messages.

3. Protocol Details

3.1. Session Initiation

At the beginning of a PA session, a PCP client and a PCP server need to exchange a series of PA messages in order to perform an EAP authentication process. Each PA message MUST contain an AUTHENTICATION Opcode and may optionally contain a set of Options for various purposes (e.g., transporting authentication messages and

session management). The opcode-specific information in a AUTHENTICATION Opcode consists of two fields : Session ID and Sequence Number. The Session ID field is used to identify the PA session to which the message belongs. The sequence number field is used to detect whether reordering or duplication occurred during message delivery.

3.1.1. Authentication triggered by the client

When a PCP client intends to proactively initiate a PA session with a PCP server, it sends a PA-Initiation message (a PA-Client message with the result code "INITIATION") to the PCP server. Section 5.1 updates the PCP request message format with result codes for the PCP Authentication mechanism. In the opcode-specific information of the message, the Session ID and Sequence Number fields are set as 0. The PA-Client message MUST also contain a NONCE option defined in Section 5.3 which consists of a random nonce.

After receiving the PA-Initiation, if the PCP server agrees to initiate a PA session with the PCP client, it will reply with a PA-Server message which contains an EAP Request and the result code field of this PA-Server message is set to AUTHENTICATION_REQUEST. In addition, the server MUST assign a unique session identifier to distinctly identify this session, and fill the identifier into the Session ID field in the opcode-specific information of the PA-Server message. The Sequence Number field of the message is set as 0. The PA-Server message MUST contain a NONCE option so as to send the nonce value back. The nonce will then be used by the PCP client to check the freshness of this message. Subsequent PCP messages within this PA session MUST contain this session identifier.

```

PCP client                                     PCP server
|-----> PA-Initiation----->
|      (Seq=0, rc=INITIATION, Session ID=0)
|
|<----- PA-Server -----<
|      (Seq=0, Session ID=X, EAP request,
|      rc=AUTHENTICATION_REQUEST)
|
|-----> PA-Client ----->
|      (Seq=1, Session ID=X, EAP response,
|      rc=AUTHENTICATION_REPLY)
|
|<----- PA-Server -----<
|      (Seq=1, Session ID=X, EAP request,
|      rc=AUTHENTICATION_REQUEST)

```

3.1.2. Authentication triggered by the server

In the scenario where a PCP server receives a common PCP request message from a PCP client which needs to be authenticated, the PCP server rejects the request with a AUTHENTICATION_REQUIRED error code and can reply with a unsolicited PA-Server message to initiate a PA session. The result code field of this PA-Server message is set to AUTHENTICATION_REQUEST. In addition, the PCP server MUST assign a Session ID for the session and transfer it within the PA-Server message. The Sequence Number field in the PA-Server message is set as 0. If the PCP client retries the common request before EAP authentication is successful then it will receive AUTHENTICATION_REQUIRED error code from the PCP server. In the PA messages exchanged afterwards in this session, the Session ID will be used in order to help session partners distinguish the messages within this session from those not within. When the PCP client receives this initial PA-Server message from the PCP server, it can reply with a PA-Client message or silently discard the request message according to its local policies. In the PA-Client message, a NONCE option which consists of a random nonce MAY be appended. If so, in the next PA-Server message, the PCP server MUST forward the nonce back within a NONCE option.

```

PCP client                                     PCP server
|-----> | Common PCP request
|-----< | Common PCP response
|         | rc=AUTHENTICATION_REQUIRED)
|-----< | PA-Server -----
|         | (Seq=0, Session ID=X, EAP request)
|         | rc=AUTHENTICATION_REQUEST)
|-----> | PA-Client -----
|         | (Seq=0, Session ID=X, EAP response)
|         | rc=AUTHENTICATION_REPLY)
|-----< | PA-Server -----
|         | (Seq=1, Session ID=X, EAP request,
|         | rc=AUTHENTICATION_REQUEST)

```

3.1.3. Authentication using EAP

In a PA session, an EAP request message is transported within a PA-Server message and an EAP response message is transported within a PA-Client message. EAP relies on the underlying protocol to provide

reliable transmission; any reordered delivery or loss of packets occurring during transportation must be detected and addressed. Therefore, after sending out a PA-Server message, the PCP server will not send a new PA-Server message in the same PA session until it receives a PA-Client message with a proper sequence number from the PCP client, and vice versa. If a PCP client receives a PA message containing an EAP request and cannot generate an EAP response immediately due to certain reasons (e.g., waiting for human input to construct a EAP message or due to EAP message fragmentation waiting for the additional PA messages in order to construct a complete EAP message), the PCP device MUST reply with a PA-Acknowledgement message (PA message with a RECEIVED_PAK Option) to indicate that the message has been received. This approach not only can avoid unnecessary retransmission of the PA message but also can guarantee the reliable message delivery in conditions where a PCP device needs to receive multiple PA messages carrying the fragmented EAP request before generating an EAP response. The number of EAP messages exchanged between the PCP client and PCP server depends on the EAP method used for authentication.

In this approach, PCP client and a PCP server MUST perform a key-generating EAP method in authentication. Particularly, a PCP authentication implementation MUST support EAP-TTLS [RFC5281] and SHOULD support TEAP [RFC7170]. Therefore, after a successful authentication procedure, a Master Session Key (MSK) will be generated. If the PCP client and the PCP server want to generate a transport key using the MSK, they need to agree upon a Pseudo-Random Function (PRF) for the transport key derivation and a MAC algorithm to provide data origin authentication for subsequent PCP messages. In order to do this, the PCP server needs to append a set of PRF Options and MAC_ALGORITHM Options to the initial PA-Server message. Each PRF Option contains a PRF that the PCP server supports, and each MAC_ALGORITHM Option contains a MAC (Message Authentication Code) algorithm that the PCP server supports. Moreover, in the first PA-Server message, the server MAY also attach an ID_INDICATOR Option defined in Section 5.11 to direct the client to choose correct credentials. After receiving the options, the PCP client MUST select the PRF and the MAC algorithm which it would like to use, and then adds the associated PRF and MAC Algorithm Options to the next PA-Client message.

After the EAP authentication, the PCP server sends out a PA-Server message to indicate the EAP authentication and PCP authorization results. If the EAP authentication succeeds, the result code of the PA-Server message is AUTHENTICATION_SUCCEEDED. In this case, before sending out the PA-Server message, the PCP server MUST update the PCP SA with the MSK and transport key, and use the derived transport key to generate a digest for the message. The digest is transported

within an PA_AUTHENTICATION_TAG Option for PCP Auth. A more detailed description of generating the authentication data can be found in Section 6.1. In addition, the PA-Server message MUST also contain a SESSION_LIFETIME Option defined in Section 5.9 which indicates the lifetime of the PA session (i.e., the lifetime of the MSK). After receiving the PA-Server message, the PCP client then needs to generate a PA-Client message as response. If the PCP client also authenticates the PCP server, the result code of the PA-Client message is AUTHENTICATION_SUCCEEDED. In addition, the PCP client needs to update the PCP SA with the MSK and transport key, and uses the derived transport key to secure the message. From then on, all the PCP messages within the session are secured with the transport key and the MAC algorithm specified in the PCP SA. The first secure PA-client message from the client MUST include the set of PRF and MAC_ALGORITHM options received from the PCP server. The PCP server determines if the set of algorithms conveyed by the client matches the set it had initially sent, to detect an algorithm downgrade attack. If the server detects a downgrade attack then it MUST send a PA-Server message with result code DOWNGRADE_ATTACK_DETECTED and terminate the session. If the PCP client sends common PCP request within the PA session without AUTHENTICATION_TAG option then the PCP server rejects the request by returning AUTHENTICATION_REQUIRED error code.

If a PCP client/server cannot authenticate its session partner, the device sends out a PA message with the result code, AUTHENTICATION_FAILED. If the EAP authentication succeeds but authorization fails, the device making the decision sends out a PA message with the result code, AUTHORIZATION_FAILED. In these two cases, after the PA message is sent out, the PA session MUST be terminated immediately. It is possible for independent PCP clients on the host to create multiple PA sessions with the PCP server.

3.2. Recovery from lost PA session

If a PCP server resets or loses the PCP SA due to reboot, power failure, or any reason then it sends unsolicited ANNOUNCE response as explained in section 14.1.3 of [RFC6887] to the PCP client. Upon receiving the ANNOUNCE response with an anomalous Epoch time, PCP client deduces that the server may have lost state. The ANNOUNCE is either bogus (an attack), legitimate, or not seen by the client. These three cases are described below:

- o PCP client sends integrity-protected unicast ANNOUNCE request to the PCP server to check if the PCP server has indeed lost the state or an attacker has sent the ANNOUNCE response.

- * If integrity-protected success response is received from the PCP server then the PCP client determines that the PCP server has not lost the PA session, and the unsolicited ANNOUNCE response was sent by an attacker.
- * If the PCP server responds to the ANNOUNCE request with UNKNOWN_SESSION_ID error code then the PCP client MUST initiate full EAP authentication with the PCP server as explained in Section 3.1.1. After EAP authentication is successful PCP client updates the PCP SA and issues new common PCP requests to recreate any lost mapping state.
- o In a scenario where the PCP server has lost the PCP SA but did not inform the PCP client, if the PCP client sends PCP request integrity-protected then the PCP server rejects the request with UNKNOWN_SESSION_ID error code. The PCP client then initiates full EAP authentication with the PCP server as explained in Section 3.1.1 and updates the PCP SA after successful authentication.

If the PCP client resets or loses the PCP SA due to reboot, power failure, or any reason and sends common PCP request then the PCP server rejects the request with AUTHENTICATION_REQUIRED error code. The PCP client MUST authenticate with the PCP server and after EAP authentication is successful retry the common PCP request with AUTHENTICATION_TAG option. The PCP server MUST update the PCP SA after successful EAP authentication.

3.3. Session Termination

A PA session can be explicitly terminated by either session partner. A PCP Server may explicitly request termination of the session by sending an unsolicited termination-indicating PA response (a PA response with a result code "SESSION-TERMINATED"). Upon receiving a termination-indicating message, the PCP client MUST respond with a termination-indicating PA message, and MUST then remove the associated PCP SA. To accommodate packet loss, the PCP server MAY transmit the termination-indicating PA response up to ten times (with an appropriate Epoch Time value in each to reflect the passage of time between transmissions) provided that the interval between the first two notifications is at least 250 ms, and the interval between subsequent notification at least doubles.

A PCP client may explicitly request termination of the session by sending a termination-indicating PA request (a PA request with a result code "SESSION-TERMINATED"). After receiving a termination-indicating message from the PCP client, a PCP server MUST respond with a termination-indicating PA response and remove the PCP SA

immediately. When the PCP client receives the termination-indicating PA response, it MUST remove the associated PCP SA immediately.

3.4. Session Re-Authentication

A session partner may select to perform EAP re-authentication if it would like to update the PCP SA without initiating a new PA session. For example a re-authentication procedure could be triggered for the following reasons:

- o The session lifetime needs to be extended.
- o The sequence number is going to reach the maximum value. Specifically, when the sequence number reaches $2^{32} - 2^{16}$, the session partner MUST trigger re-authentication.

When the PCP server would like to initiate a re-authentication, it sends the PCP client a PA-Server message. The result code of the message is set to "RE-AUTHENTICATION", which indicates the message is for a re-authentication process. If the PCP client would like to start the re-authentication, it will send a PA-Client message to the PCP server, with the result code of the PA-Client message set to "RE-AUTHENTICATION". Then, the session partners exchange PA messages to transfer EAP messages for the re-authentication. During the re-authentication procedure, the session partners protect the integrity of PA messages with the key and MAC algorithm specified in the current PCP SA; the sequence numbers associated with the message will continue to keep increasing according to Section 6.3. The result code for PA-Server message carrying EAP request will be set to AUTHENTICATION_REQUIRED and PA-Client message carrying EAP response will be set to AUTHENTICATION_REPLY.

If the EAP re-authentication succeeds, the result code of the last PA-Server message is "AUTHENTICATION_SUCCEEDED". In this case, before sending out the PA-Server message, the PCP server MUST update the SA and use the new key to generate a digest for the PA-Server message and subsequent PCP messages. In addition, the PA-Server message MUST be appended with a SESSION_LIFETIME Option which indicates the new lifetime of the PA session. PA and PCP message sequence numbers must also be reset to zero.

If the EAP authentication fails, the result code of the last PA-Server message is "AUTHENTICATION_FAILED". If the EAP authentication succeeds but authorization fails, the result code of the last PA-Server message is "AUTHORIZATION_FAILED". In the latter two cases, the PA session MUST be terminated immediately after the last PA message exchange. If for some unknown reason re-authentication is

not performed and session lifetime has expired then PA session MUST be terminated immediately.

During re-authentication, the session partners can also exchange common PCP messages in parallel. The common PCP messages MUST be protected with the current SA until the new SA has been generated. The sequence of EAP messages exchanged for re-authentication will not change, regardless of the PCP device triggering re-authentication. If the PCP server receives re-authentication request from the PCP client after it had signaled re-authentication request then it should discard its request and respond to the re-authentication request from the PCP client.

4. PA Security Association

At the beginning of a new PA session, each PCP device must create and initialize state information for a new PA Security Association (PCP SA) to maintain its state information for the duration of the PA session. The parameters of a PCP SA are listed as follows:

- o IP address and UDP port number of the PCP client
- o IP address and UDP port number of the PCP server
- o Session Identifier
- o Sequence number for the next outgoing PA message
- o Sequence number for the next incoming PA message
- o Sequence number for the next outgoing common PCP message
- o Sequence number for the next incoming common PCP message
- o Last outgoing message payload
- o Retransmission interval
- o The master session key (MSK) generated by the EAP method.
- o The MAC algorithm that the transport key should use to generate digests for PCP messages.
- o The pseudo random function negotiated in the initial PA-Server and PA-Client message exchange for the transport key derivation
- o The transport key derived from the MSK to provide integrity protection and data origin authentication for the messages in the

PA session. The lifetime of the transport key SHOULD be identical to the lifetime of the session.

- o The nonce selected by the PCP client at the initiation of the session.
- o The Key ID associated with Transport key.

Particularly, the transport key is computed in the following way: Transport key = prf(MSK, "IETF PCP" || Session ID || Nonce || key ID), where:

- o prf: The pseudo-random function assigned in the Pseudo-random function parameter.
- o MSK: The master session key generated by the EAP method.
- o "IETF PCP": The ASCII code representation of the non-NULL terminated string (excluding the double quotes around it).
- o '||' : is the concatenation operator.
- o Session ID: The ID of the session which the MSK is derived from.
- o Nonce: The nonce selected by the client and transported in the Initial PA-Client message.
- o Key ID: The ID assigned for the transport key.

5. Packet Format

5.1. Packet Format of PCP Auth Messages

The format of the PA-Server message is identical to the response message format specified in Section 7.2 of [RFC6887]. The result code for PA-Sever message carrying EAP request MUST be set to AUTHENTICATION_REQUEST.

As illustrated in Figure 1, this document updates the reserved field in the request header specified in Section 7.1 of [RFC6887] to carry Opcode-specific data.

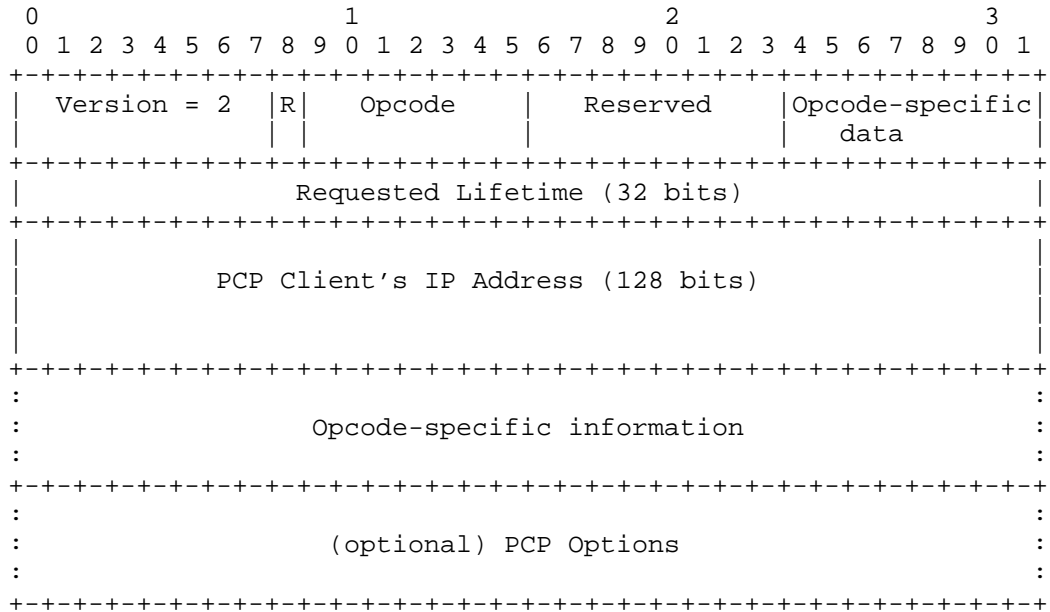


Figure 1. Request Packet Format

As illustrated in Figure 2, the PA-Client messages use the request header specified in Figure 1. The Opcode-specific data is used to transfer the result codes (e.g., "INITIATION", "AUTHENTICATION_FAILED"). Other fields in Figure 2 are described in Section 7.1 of [RFC6887]. The result code for PA-Client message carrying EAP response MUST be set to AUTHENTICATION_REPLY.

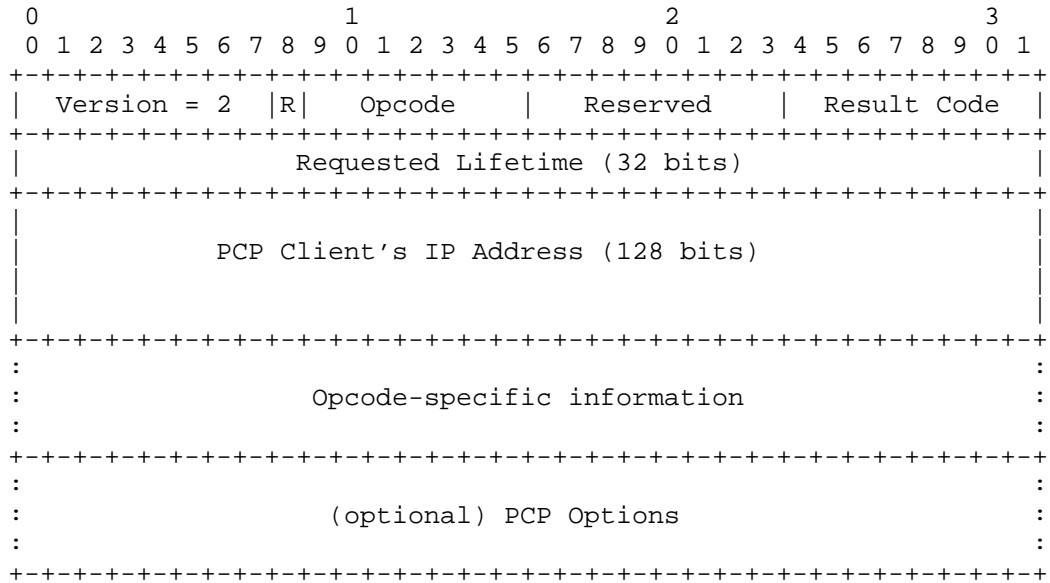
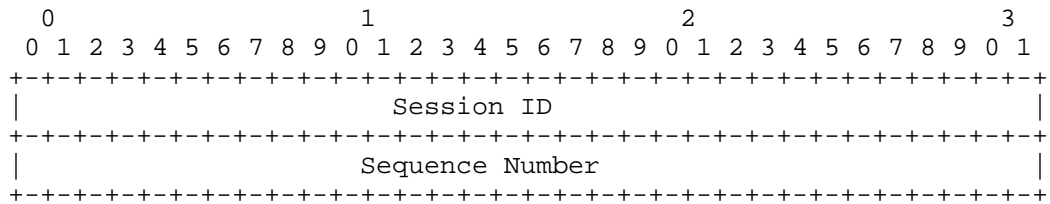


Figure 2. PA-Client message Format

The Requested Lifetime field of PA-Client message and Lifetime field of PA-Server message are both set to 0 on transmission and ignored on reception.

5.2. Opcode-specific information of AUTHENTICATION Opcode

The following diagram shows the format of the Opcode-specific information for the AUTHENTICATION Opcode.

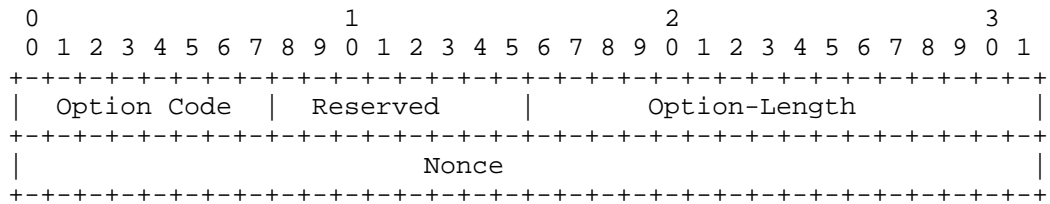


Session ID: This field contains a 32-bit PA session identifier.

Sequence Number: This field contains a 32-bit sequence number. A sequence number needs to be incremented on every new (non-retransmission) outgoing PA message in order to provide an ordering guarantee for PA messages.

5.3. NONCE Option

Because the session identifier of a PA session is determined by the PCP server, a PCP client does not know the session identifier which will be used when it sends out a PA-Initiation message. In order to prevent an attacker from interrupting the authentication process by sending off-line generated PA-Server messages, the PCP client needs to generate a random number as a nonce in the PA-Initiation message. The PCP server will append the nonce within the initial PA-Server message. If the PA-Server message does not carry the correct nonce, the message MUST be discarded silently.



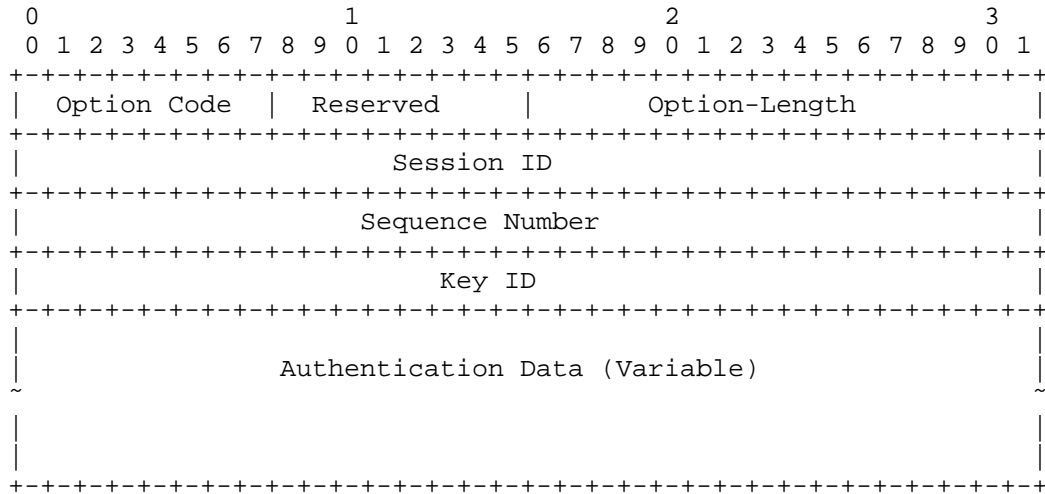
Option Code: TBA-130.

Reserved: 8 bits. MUST be set to 0 on transmission and MUST be ignored on reception.

Option-Length: 4 octets.

Nonce: A random 32 bit number which is transported within a PA-Initiation message and the corresponding reply message from the PCP server.

5.4. AUTHENTICATION_TAG Option



Because there is no authentication Opcode in common PCP messages, the authentication tag for common PCP messages needs to carry the Session ID and Sequence Number.

Option Code: TBA-131.

Reserved: 8 bits. MUST be set to 0 on transmission and MUST be ignored on reception.

Option-Length: The length of the AUTHENTICATION_TAG Option for Common PCP message (in octets), including the 12 octet fixed header and the variable length of the authentication data.

Session ID: A 32-bit field used to identify the session to which the message belongs and identify the secret key used to create the message digest appended to the PCP message.

Sequence Number: A 32-bit sequence number. In this solution, a sequence number needs to be incremented on every new (non-retransmission) outgoing common PCP message in order to provide ordering guarantee for common PCP messages.

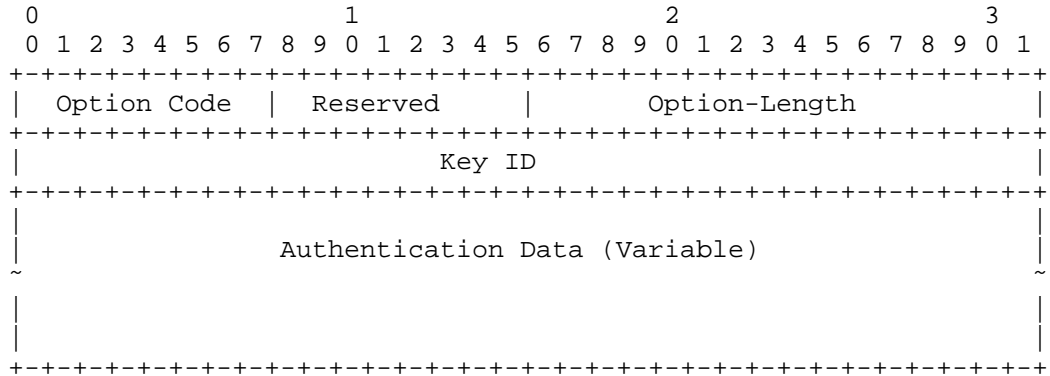
Key ID: The ID associated with the transport key used to generate authentication data. This field is filled with zero if the MSK is directly used to secure the message.

Authentication Data: A variable-length field that carries the Message Authentication Code for the Common PCP message. The generation of the digest varies according to the algorithms

specified in different PCP SAs. This field MUST end on a 32-bit boundary, padded with 0's when necessary.

5.5. PA_AUTHENTICATION_TAG option

This option is used to provide message authentication for PA messages. Compared with the AUTHENTICATION_TAG Option for Common PCP Messages, the Session ID field and the Sequence Number field are removed because such information is provided in the Opcode-specific information of AUTHENTICATION Opcode.



Option Code: TBA-132.

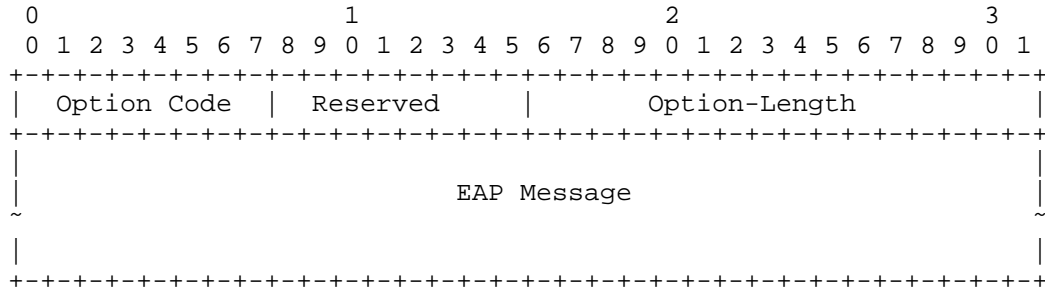
Reserved: 8 bits. MUST be set to 0 on transmission and MUST be ignored on reception.

Option-Length: The length of the PA_AUTHENTICATION Option for PCP Auth message (in octet), including the 4 octet fixed header and the variable length of the authentication data.

Key ID: The ID associated with the transport key used to generate authentication data. This field is filled with zero if the MSK is directly used to secure the message.

Authentication Data: A variable-length field that carries the Message Authentication Code for the PCP Auth message. The generation of the digest varies according to the algorithms specified in different PCP SAs. This field MUST end on a 32-bit boundary, padded with null characters when necessary.

5.6. EAP_PAYLOAD Option



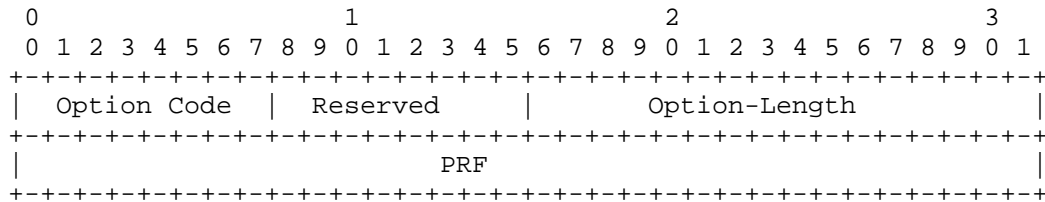
Option Code: TBA-133.

Reserved: 8 bits. MUST be set to 0 on transmission and MUST be ignored on reception.

Option-Length: Variable

EAP Message: The EAP message transferred. Note this field MUST end on a 32-bit boundary, padded with 0's when necessary.

5.7. PRF Option



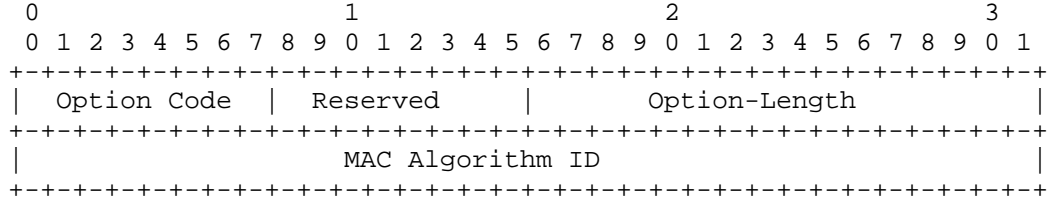
Option Code: TBA-134.

Reserved: 8 bits. MUST be set to 0 on transmission and MUST be ignored on reception.

Option-Length: 4 octets.

PRF: The Pseudo-Random Function which the sender supports to generate an MSK. This field contains an IKEv2 Transform ID of Transform Type 2 [RFC7296][RFC4868]. A PCP implementation MUST support PRF_HMAC_SHA2_256 (5).

5.8. MAC_ALGORITHM Option



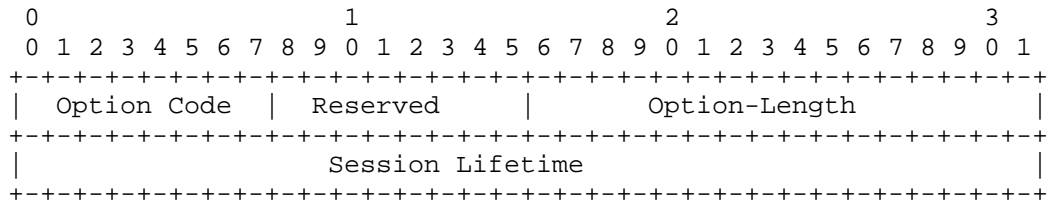
Option Code: TBA-135.

Reserved: 8 bits. MUST be set to 0 on transmission and MUST be ignored on reception.

Option-Length: 4 octets.

MAC Algorithm ID: Indicate the MAC algorithm which the sender supports to generate authentication data. The MAC Algorithm ID field contains an IKEv2 Transform ID of Transform Type 3 [RFC7296][RFC4868]. A PCP implementation MUST support AUTH_HMAC_SHA2_256_128 (12).

5.9. SESSION_LIFETIME Option



Option Code: TBA-136.

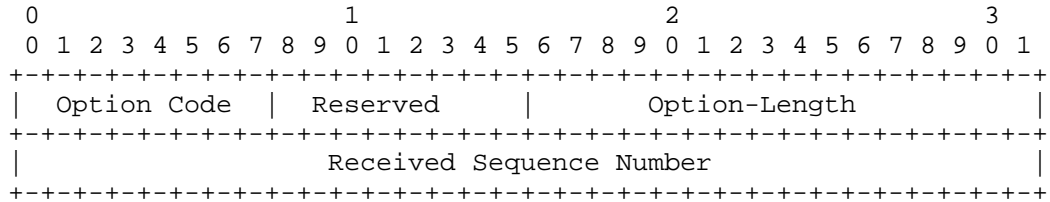
Reserved: 8 bits. MUST be set to 0 on transmission and MUST be ignored on reception.

Option-Length: 4 octets.

Session Lifetime: An unsigned 32-bit integer, in seconds, ranging from 0 to 2^32-1 seconds. The lifetime of the PA Session, which is decided by the authorization result.

5.10. RECEIVED_PAK Option

This option is used in a PA-Acknowledgement message to indicate that a PA message with the contained sequence number has been received.



Option Code: TBA-137.

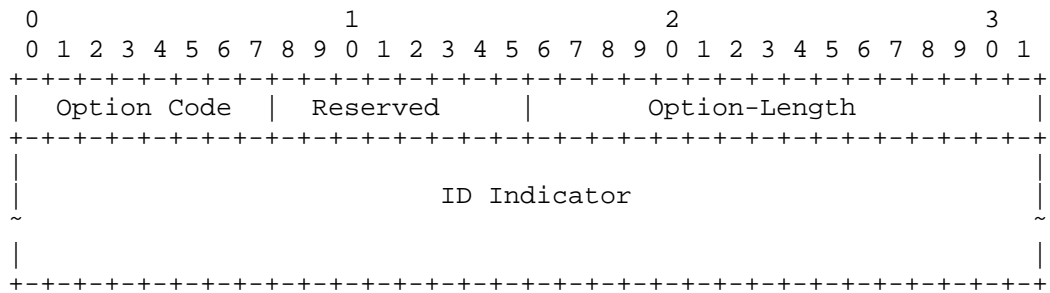
Reserved: 8 bits. MUST be set to 0 on transmission and MUST be ignored on reception.

Option-Length: 4 octets.

Received Sequence Number: The sequence number of the last received PA message.

5.11. ID_INDICATOR Option

The ID_INDICATOR option is used by the PCP client to determine which credentials to provide to the PCP server.



Option Code: TBA-138.

Reserved: 8 bits. MUST be set to 0 on transmission and MUST be ignored on reception.

Option-Length: Variable.

ID Indicator: The identity of the authority that issued the EAP credentials to be used to authenticate the client. The field MUST

NOT be null terminated and its length is indicated by the Option-Length field. In particular when a client receives a ID_INDICATOR option, it MUST NOT rely on the presence of a NUL character in the wire format data to identify the end of the ID Indicator field.

The field MUST end on a 32-bit boundary, padded with 0's when necessary. The ID indicator field is UTF-8 encoded [RFC3629] Unicode string conforming to the "UsernameCaseMapped" profile of the PRECIS IdentifierClass [I-D.ietf-precis-saslprepbis]. The PCP client validates that the ID indicator field conforms to the "UsernameCaseMapped" profile of the PRECIS IdentifierClass. The PCP client enforces the rules specified in section 3.2.2 of [I-D.ietf-precis-saslprepbis] to map the ID indicator field. The PCP client compares the resulting string with the ID indicators stored locally on the PCP client to pick the credentials for authentication. The two indicator strings are to be considered equivalent by the client if and only if they are an exact octet-for-octet match.

6. Processing Rules

6.1. Authentication Data Generation

After successful EAP authentication process, every subsequent PCP message within the PA session MUST carry an authentication tag which contains the digest of the PCP message for data origin authentication and integrity protection.

- o Before generating a digest for a PA message, a device needs to first locate the PCP SA according to the session identifier and then get the transport key. Then the device appends an PA_AUTHENTICATION_TAG Option for PCP Auth at the end of the PCP Auth message. The length of the Authentication Data field is decided by the MAC algorithm adopted in the session. The device then fills the Key ID field with the key ID of the transport key, and sets the Authentication Data field to 0. After this, the device generates a digest for the entire PCP message (including the PCP header and PA_AUTHENTICATION_TAG Option) using the transport key and the associated MAC algorithm, and inserts the generated digest into the Authentication Data field.
- o Similar to generating a digest for a PA message, before generating a digest for a common PCP message, a device needs to first locate the PCP SA according to the session identifier and then get the transport key. Then the device appends the AUTHENTICATION_TAG Option at the end of common PCP message. The length of the Authentication Data field is decided by the MAC algorithm adopted in the session. The device then uses the corresponding values

derived from the SA to fill the Session ID field, the Sequence Number field and the Key ID field, and sets the Authentication Data field to 0. After this, the device generates a digest for the entire PCP message (including the PCP header and AUTHENTICATION_TAG Option) using the transport key and the associated MAC algorithm, and inputs the generated digest into the Authentication Data field.

6.2. Authentication Data Validation

When a device receives a common PCP message with an AUTHENTICATION_TAG Option for Common PCP Messages, the device needs to use the Session ID transported in the option to locate the proper SA, and then find the associated transport key (using the key ID in the option) and the MAC algorithm. If no proper SA or transport key is found or the sequence number is invalid (see Section 6.5), the PCP device stops processing the PCP message and discards the message silently. After storing the value of the Authentication field of the AUTHENTICATION_TAG Option, the device fills the Authentication field with zeros. Then, the device generates a digest for the message (including the PCP header and Authentication Tag Option) with the transport key and the MAC algorithm. If the value of the newly generated digest is identical to the stored one, the device can ensure that the message has not been tampered with, and the validation succeeds. Otherwise, the PCP device stops processing the PCP message and silently discards the message.

Similarly, when a device receives a PA message with an PA_AUTHENTICATION_TAG Option for PCP Authentication, the device needs to use the Session ID transported in the Opcode to locate the proper SA, and then find the associated transport key (using the key ID in the option) and the MAC algorithm. If no proper SA or transport key is found or the sequence number is invalid (see Section 6.4), the PCP device stops processing the PCP message and discards the message. After storing the value of the Authentication field of the PA_AUTHENTICATION_TAG Option, the device fills the Authentication field with zeros. Then, the device generates a digest for the message (including the PCP header and PA_AUTHENTICATION_TAG Option) with the transport key and the MAC algorithm. If the value of the newly generated digest is identical to the stored one, the device can ensure that the message has not been tampered with, and the validation succeeds. Otherwise, the PCP device stops processing the PCP message and silently discards the message.

6.3. Retransmission Policies for PA Messages

Because EAP relies on the underlying protocols to provide reliable transmission, after sending a PA message, a PCP client/server MUST NOT send out any subsequent messages until receiving a PA message with a proper sequence number from the peer. If no such a message is received the PCP device will re-send the last message according to retransmission policies. This work reuses the retransmission policies specified in the base PCP protocol (Section 8.1.1 of [RFC6887]). In the base PCP protocol, such retransmission policies are only applied by PCP clients. However, in this work, such retransmission policies are also applied by the PCP servers. If Maximum retransmission duration seconds have elapsed and no expected response is received, the device will terminate the session and discard the current SA.

As illustrated in Section 3.1.3, in order to avoid unnecessary retransmission, the device receiving a PA message MUST send a PA-Acknowledgement message to the sender of the PA message when it cannot send a PA response immediately. The PA-Acknowledgement message is used to indicate the receipt of the PA message. When the sender receives the PA-Acknowledgement message, it will stop the retransmission.

Note that the last PA messages transported within the phases of session initiation, session re-authentication, and session termination do not have to follow the above policies since the devices sending out those messages do not expect any further PA messages.

When a device receives a re-transmitted last incoming PA message from its session partner, it MUST try to answer it by sending the last outgoing PA message again. However, if the duplicate message has the same sequence number but is not bit-wise identical to the original message then the device MUST discard it. In order to achieve this function, the device may need to maintain the last incoming and the associated outgoing messages. In this case, if no outgoing PA message has been generated for the received duplicate PA message yet, the device needs to send a PA-Acknowledgement message. The rate of replying to duplicate PA messages MUST be limited to provide robustness against denial of service (DoS) attacks. The details of rate limiting are outside the scope of this specification.

6.4. Sequence Numbers for PCP Auth Messages

PCP uses UDP to transport signaling messages. As an un-reliable transport protocol, UDP does not guarantee ordered packet delivery and does not provide any protection from packet loss. In order to

ensure the EAP messages are exchanged in a reliable way, every PCP message exchanged during EAP authentication must carry a monotonically increasing sequence number. During a PA session, a PCP device needs to maintain two sequence numbers for PA messages, one for incoming PA messages and one for outgoing PA messages. When generating an outgoing PA message, the device adds the associated outgoing sequence number to the message and increments the sequence number maintained in the SA by 1. When receiving a PA message from its session partner, the device will not accept it if the sequence number carried in the message does not match the incoming sequence number the device maintains. After confirming that the received message is valid, the device increments the incoming sequence number maintained in the SA by 1.

The above rules are not applicable to PA-Acknowledgement messages (i.e., PA messages containing a RECEIVED_PAK Option). A PA-Acknowledgement message does not transport any EAP message and only indicates that a PA message is received. Therefore, reliable transmission of PA-Acknowledgement messages is not required. For instance, after sending out a PA-Acknowledgement message, a device generates an EAP response. In this case, the device need not have to confirm whether the PA-Acknowledgement message has been received by its session partner or not. Therefore, when receiving or sending out a PA-Acknowledgement message, the device MUST NOT increase the corresponding sequence number stored in the SA. Otherwise, loss of a PA-Acknowledgement message will cause a mismatch in sequence numbers.

Another exception is the message retransmission scenario. As discussed in Section 6.3, when a PCP device does not receive any response from its session partner it needs to retransmit the last outgoing PA message following the retransmission procedure specified in section 8.1.1 of [RFC6887]. The original message and duplicate messages MUST be bit-wise identical. When the device receives such a duplicate PA message from its session partner, it MUST send the last outgoing PA message again. In such cases, the maintained incoming and outgoing sequence numbers will not be affected by the message retransmission.

6.5. Sequence Numbers for Common PCP Messages

When transporting common PCP messages within a PA session, a PCP device needs to maintain a sequence number for outgoing common PCP messages and a sequence number for incoming common PCP messages. When generating a new outgoing PCP message, the PCP device updates the Sequence Number field in the AUTHENTICATION_TAG option with the outgoing sequence number maintained in the SA and increments the outgoing sequence number by 1.

When receiving a PCP message from its session partner, the PCP device will not accept it if the sequence number carried in the message is smaller than the incoming sequence number the device maintains. This approach can protect the PCP device from replay attacks. After confirming that the received message is valid, the PCP device will update the incoming sequence number maintained in the PCP SA with the sequence number of the incoming message.

Note that the sequence number in the incoming message may not exactly match the incoming sequence number maintained locally. As discussed in the base PCP specification [RFC6887], if a PCP client is no longer interested in the PCP transaction and has not yet received a PCP response from the server then it will stop retransmitting the PCP request. After that, the PCP client might generate new PCP requests for other purposes using the current SA. In this case, the sequence number in the new request will be larger than the sequence number in the old request and so will be larger than the incoming sequence number maintained in the PCP server.

Note that in the base PCP specification [RFC6887], a PCP client needs to select a nonce in each MAP or PEER request, and the nonce is sent back in the response. However, it is possible for a client to use the same nonce in multiple MAP or PEER requests, and this may cause a potential risk of replay attacks. This attack is addressed by using the sequence number in the PCP response.

6.6. MTU Considerations

EAP methods are responsible for MTU handling, so no special facilities are required in PCP to deal with MTU issues. Particularly, EAP lower layers indicate to EAP methods and AAA servers the MTU of the lower layer. EAP methods such as EAP-TLS [RFC5216], TEAP [RFC7170], and others that are likely to exceed reasonable MTUs provide support for fragmentation and reassembly. Others, such as EAP-GPSK [RFC5433] assume they will never send packets larger than the MTU and use small EAP packets.

If an EAP message is too long to be transported within a single PA message, it will be divided into multiple sections and sent within different PA messages. Note that the receiver may not be able to know what to do in the next step until it has received all the sections and reconstructed the complete EAP message. In this case, in order to guarantee reliable message transmission, after receiving a PA message, the receiver replies with a PA-Acknowledgement message to notify the sender to send the next PA message.

7. IANA Considerations

The following PCP Opcode is to be allocated in the mandatory-to-process range from the standards action range (the registry for PCP Opcodes is maintained in <http://www.iana.org/assignments/pcp-parameters>):

TBA AUTHENTICATION Opcode.

The following PCP result codes are to be allocated in the mandatory-to-process range from the standards action range (the registry for PCP result codes is maintained in <http://www.iana.org/assignments/pcp-parameters>):

TBA INITIATION: The client indication to the server for authentication.

TBA AUTHENTICATION_REQUIRED: The error response is signaled to the client that EAP authentication is required.

TBA AUTHENTICATION_FAILED: This error response is signaled to the client if EAP authentication had failed.

TBA AUTHENTICATION_SUCCEEDED: This success response is signaled to the client if EAP authentication had succeeded.

TBA AUTHORIZATION_FAILED: This error response is signaled to the client if the EAP authentication had succeeded but authorization failed.

TBA SESSION_TERMINATED: This PCP result code indicates to the partner that the PA session must be terminated.

TBA UNKNOWN_SESSION_ID: The error response is signaled from the PCP server that there is no known PA session associated with the Session ID signaled in the PA request or common PCP request from the PCP client.

TBA DOWNGRADE_ATTACK_DETECTED: This error response is signaled to the client if the server detects downgrade attack.

TBA AUTHENTICATION_REQUEST: The server indication to the client that EAP request is signaled in the PA message.

TBA AUTHENTICATION_REPLY: The client indication to the server that EAP response is signaled in the PA message.

The following PCP Option Codes are to be allocated in the mandatory-to-process range from the standards action range (the registry for PCP Options is maintained in <http://www.iana.org/assignments/pcp-parameters>):

7.1. NONCE

Option Name: NONCE

option-code: TBA-130 in the mandatory-to-process range (IANA).

Purpose: See Section 5.3.

Valid for Opcodes: Authentication Opcode.

option-len: Option Length is 4 octets.

May appear in: request and response.

Maximum occurrences: 1.

7.2. AUTHENTICATION_TAG

Option Name: AUTHENTICATION_TAG

option-code: TBA-131 in the mandatory-to-process range (IANA).

Purpose: See Section 5.4.

Valid for Opcodes: MAP, PEER and ANNOUNCE Opcodes.

option-len: Variable length.

May appear in: request and response.

Maximum occurrences: 1.

7.3. PA_AUTHENTICATION_TAG

Option Name: PA_AUTHENTICATION_TAG

option-code: TBA-132 in the mandatory-to-process range (IANA).

Purpose: See Section 5.5.

Valid for Opcodes: Authentication Opcode.

option-len: Variable length.

May appear in: request and response.

Maximum occurrences: 1.

7.4. EAP_PAYLOAD

Option Name: EAP_PAYLOAD.

option-code: TBA-133 in the mandatory-to-process range (IANA).

Purpose: See Section 5.6.

Valid for Opcodes: Authentication Opcode.

option-len: Variable length.

May appear in: request and response.

Maximum occurrences: 1.

7.5. PRF

Option Name: PRF.

option-code: TBA-134 in the mandatory-to-process range (IANA).

Purpose: See Section 5.7.

Valid for Opcodes: Authentication Opcode.

option-len: Option Length is 4 octets.

May appear in: request and response.

Maximum occurrences: as many as fit within maximum PCP message size.

7.6. MAC_ALGORITHM

Option Name: MAC_ALGORITHM.

option-code: TBA-135 in the mandatory-to-process range (IANA).

Purpose: See Section 5.8.

Valid for Opcodes: Authentication Opcode.

option-len: Option Length is 4 octets.

May appear in: request and response.

Maximum occurrences: as many as fit within maximum PCP message size.

7.7. SESSION_LIFETIME

Option Name: SESSION_LIFETIME.

option-code: TBA-136 in the mandatory-to-process range (IANA).

Purpose: See Section 5.9.

Valid for Opcodes: Authentication Opcode.

option-len: Option Length is 4 octets.

May appear in: response.

Maximum occurrences: 1.

7.8. RECEIVED_PAK

Option Name: RECEIVED_PAK.

option-code: TBA-137 in the mandatory-to-process range (IANA).

Purpose: See Section 5.10.

Valid for Opcodes: Authentication Opcode.

option-len: Option Length is 4 octets.

May appear in: request and response.

Maximum occurrences: 1.

7.9. ID_INDICATOR

Option Name: ID_INDICATOR.

option-code: TBA-138 in the mandatory-to-process range (IANA).

Purpose: See Section 5.11.

Valid for Opcodes: Authentication Opcode.

option-len: Variable length.

May appear in: response.

Maximum occurrences: 1.

8. Security Considerations

In this work, after a successful EAP authentication process is performed between two PCP devices, an MSK will be exported. The MSK will be used to derive the transport keys to generate MAC digests for subsequent PCP message exchanges. However, before a transport key has been generated, the PA messages exchanged within a PA session have little cryptographic protection, and if there is no already established security channel between two session partners, these messages are subject to man-in-the-middle attacks and DOS attacks. For instance, the initial PA-Server and PA-Client message exchange is vulnerable to spoofing attacks as these messages are not authenticated and integrity protected. In addition, because the PRF and MAC algorithms are transported at this stage, an attacker may try to remove the PRF and MAC options containing strong algorithms from the initial PA-Server message and force the client choose the weakest algorithms. Therefore, the server needs to guarantee that all the PRF and MAC algorithms it provides support for are strong enough.

In order to prevent very basic DOS attacks, a PCP device SHOULD generate state information as little as possible in the initial PA-Server and PA-Client message exchanges. The choice of EAP method is also very important. The selected EAP method must be resilient to the attacks possible in an insecure network environment, provide user-identity confidentiality, protection against dictionary attacks, and support session-key establishment.

When a PCP proxy [I-D.ietf-pcp-proxy] is located between a PCP server and PCP clients, the proxy may perform authentication with the PCP server before it processes requests from the clients. In addition, re-authentication between the PCP proxy and PCP server will not interrupt the service that the proxy provides to the clients since the proxy is still allowed to send common PCP messages to the PCP server during that period.

9. Acknowledgements

Thanks to Dan Wing, Prashanth Patil, Dave Thaler, Peter Saint-Andre, Carlos Pignataro, Brian Haberman, Paul Kyzivat, Jouni Korhonen, Stephen Farrell and Terry Manderson for the valuable comments.

10. Change Log

[Note: This section should be removed by the RFC Editor upon publication]

10.1. Changes from wasserman-pcp-authentication-02 to ietf-pcp-authentication-00

- o Added discussion of in-band and out-of-band key management options, leaving choice open for later WG decision.
- o Removed support for fragmenting EAP messages, as that is handled by EAP methods.

10.2. Changes from wasserman-pcp-authentication-01 to -02

- o Add a nonce into the first two exchanged PCP-Auth message between the PCP client and PCP server. When a PCP client initiate the session, it can use the nonce to detect offline attacks.
- o Add the key ID field into the authentication tag option so that a MSK can generate multiple transport keys.
- o Specify that when a PCP device receives a PCP-Auth-Server or a PCP-Auth-Client message from its partner the PCP device needs to reply with a PCP-Auth-Acknowledge message to indicate that the message has been received.
- o Add the support of fragmenting EAP messages.

10.3. Changes from ietf-pcp-authentication-00 to -01

- o Editorial changes, added use cases to introduction.

10.4. Changes from ietf-pcp-authentication-01 to -02

- o Add the support of re-authentication initiated by PCP server.
- o Specify that when a PCP device receives a PCP-Auth-Server or a PCP-Auth-Client message from its partner the PCP device MAY reply with a PCP-Auth-Acknowledge message to indicate that the message has been received.
- o Discuss the format of the PCP-Auth-Acknowledge message.
- o Remove the redundant information from the Auth Opcode, and specify new result codes transported in PCP packet headers

- o
- 10.5. Changes from ietf-pcp-authentication-02 to -03
 - o Change the name "PCP-Auth-Request" to "PCP-Auth-Server"
 - o Change the name "PCP-Auth-Response" to "PCP-Auth-Client"
 - o Specify two new sequence numbers for common PCP messages in the PCP SA, and describe how to use them
 - o Specify a Authentication Tag Option for PCP Common Messages
 - o Introduce the scenario where a EAP message has to be divided into multiple sections and transported in different PCP-Auth messages (for the reasons of MTU), and introduce how to use PCP-Auth-Acknowledge messages to ensure reliable packet delivery in this case.
- 10.6. Changes from ietf-pcp-authentication-03 to -04
 - o Change the name "PCP-Auth" to "PA".
 - o Refine the retransmission policies.
 - o Add more discussion about the sequence number management .
 - o Provide the discussion about how to instruct a PCP client to choose proper credential during authentication, and an ID Indicator Option is defined for that purpose.
- 10.7. Changes from ietf-pcp-authentication-04 to -05
 - o Add contents in IANA considerations.
 - o Add discussions in fragmentation.
 - o Refine the PA messages retransmission policies.
 - o Add IANA considerations.
- 10.8. Changes from ietf-pcp-authentication-05 to -06
 - o Added mechanism to handle algorithm downgrade attack.
 - o Updated Security Considerations section.
 - o Updated ID Indicator Option.

11. References

11.1. Normative References

- [I-D.ietf-pcp-proxy]
Perreault, S., Boucadair, M., Penno, R., Wing, D., and S. Cheshire, "Port Control Protocol (PCP) Proxy Function", draft-ietf-pcp-proxy-09 (work in progress), July 2015.
- [I-D.ietf-precis-saslprepbis]
Saint-Andre, P. and A. Melnikov, "Preparation, Enforcement, and Comparison of Internationalized Strings Representing Usernames and Passwords", draft-ietf-precis-saslprepbis-18 (work in progress), May 2015.
- [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>>.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <<http://www.rfc-editor.org/info/rfc3629>>.
- [RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowitz, Ed., "Extensible Authentication Protocol (EAP)", RFC 3748, DOI 10.17487/RFC3748, June 2004, <<http://www.rfc-editor.org/info/rfc3748>>.
- [RFC4868] Kelly, S. and S. Frankel, "Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec", RFC 4868, DOI 10.17487/RFC4868, May 2007, <<http://www.rfc-editor.org/info/rfc4868>>.
- [RFC5281] Funk, P. and S. Blake-Wilson, "Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TLSv0)", RFC 5281, DOI 10.17487/RFC5281, August 2008, <<http://www.rfc-editor.org/info/rfc5281>>.
- [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>>.

- [RFC7170] Zhou, H., Cam-Winget, N., Salowey, J., and S. Hanna, "Tunnel Extensible Authentication Protocol (TEAP) Version 1", RFC 7170, DOI 10.17487/RFC7170, May 2014, <<http://www.rfc-editor.org/info/rfc7170>>.
- [RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <<http://www.rfc-editor.org/info/rfc7296>>.

11.2. Informative References

- [RFC5216] Simon, D., Aboba, B., and R. Hurst, "The EAP-TLS Authentication Protocol", RFC 5216, DOI 10.17487/RFC5216, March 2008, <<http://www.rfc-editor.org/info/rfc5216>>.
- [RFC5433] Clancy, T. and H. Tschofenig, "Extensible Authentication Protocol - Generalized Pre-Shared Key (EAP-GPSK) Method", RFC 5433, DOI 10.17487/RFC5433, February 2009, <<http://www.rfc-editor.org/info/rfc5433>>.
- [RFC5448] Arkko, J., Lehtovirta, V., and P. Eronen, "Improved Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA')", RFC 5448, DOI 10.17487/RFC5448, May 2009, <<http://www.rfc-editor.org/info/rfc5448>>.

Authors' Addresses

Margaret Wasserman
Painless Security
356 Abbott Street
North Andover, MA 01845
USA

Phone: +1 781 405 7464
Email: mrw@painless-security.com
URI: <http://www.painless-security.com>

Sam Hartman
Painless Security
356 Abbott Street
North Andover, MA 01845
USA

Email: hartmans@painless-security.com
URI: <http://www.painless-security.com>

Dacheng Zhang
Huawei
Beijing
China

Email: zhang_dacheng@hotmail.com

Tirumaleswar Reddy
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marathalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: tireddy@cisco.com

PCP
Internet-Draft
Intended status: Standards Track
Expires: November 19, 2015

T. Reddy
P. Patil
Cisco
M. Isomaki
Nokia
D. Wing
Cisco
May 18, 2015

Optimizing NAT and Firewall Keepalives Using Port Control Protocol (PCP)
draft-ietf-pcp-optimize-keepalives-06

Abstract

This document describes how Port Control Protocol is useful in reducing NAT and firewall keepalive messages for a variety of applications.

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 November 19, 2015.

Copyright Notice

Copyright (c) 2015 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

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Notational Conventions	3
3. Overview of Operation	3
3.1. Application Scenarios	3
3.2. NAT Topologies and Detection	5
3.2.1. PCP based detection	5
3.2.2. Application based detection	6
3.3. Detection of PCP unaware firewalls	6
3.4. Keepalive Optimization	7
4. Keepalive Interval Determination Procedure when PCP unaware Firewall or NAT is detected	8
5. Application-Specific Operation	9
5.1. SIP	9
5.2. HTTP	10
5.3. Media and data channels with ICE	11
5.4. Detecting Flow Failure	11
5.5. Firewalls	12
5.5.1. IPv6 Network with Firewalls	12
5.5.2. Mobile Network with Firewalls	12
6. IANA Considerations	12
7. Security Considerations	13
8. Acknowledgements	13
9. References	13
9.1. Normative References	13
9.2. Informative References	14
Appendix A. Example PHP script	14
Appendix B. Savings with PCP	15
Authors' Addresses	17

1. Introduction

Many types of applications need to keep their Network Address Translator (NAT) and Firewall (FW) mappings alive for long periods of time, even when they are otherwise not sending or receiving any traffic. This is typically done by sending periodic keep-alive messages just to prevent the mappings from expiring. As NAT/FW mapping timers may be short and unknown to the endpoint, the frequency of these keepalives may be high. An IPv4 or IPv6 host can use the Port Control Protocol (PCP)[RFC6887] to flexibly manage the IP address and port mapping information on NATs and Firewalls to facilitate communications with remote hosts. This document describes how PCP can be used to reduce keepalive messages for both client-server and peer-to-peer type of communication.

The mechanism described in this document is especially useful in cellular mobile networks, where frequent keepalive messages make the radio transition between active and power-save states causing congestion in the signaling path. The excessive time spent on the active state due to keepalives also greatly reduces the battery life of the cellular connected devices such as smartphones or tablets. [I-D.ietf-v6ops-mobile-device-profile] recommends cellular hosts to be PCP-compliant in order to save battery consumption exacerbated by keepalive messages.

2. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

This note uses terminology defined in [RFC5245] and [RFC6887].

3. Overview of Operation

3.1. Application Scenarios

PCP can help both client-server and peer-to-peer applications to reduce their keepalive rate. The relevant applications are the ones that need to keep their NAT/FW mappings alive for long periods of time, for instance to be able to send or receive application messages in both directions at any time.

A typical client-server scenario is depicted in Figure 1. A client, who may reside behind one or multiple layers of NATs/FWs, opens a connection to a globally reachable server, and keeps it open to be able to receive messages from the server at any time. The connection may be a connection-oriented transport protocol such as TCP or SCTP or connection-less transport protocol such as UDP. Protocols operating in this manner include the Session Initiation Protocol (SIP) [RFC3261], the Extensible Messaging and Presence Protocol (XMPP) [RFC3921], the Internet Mail Application Protocol (IMAP) [RFC2177] with its IDLE command, the WebSocket protocol [RFC6455] and the various HTTP long-polling protocols. There are also a number of proprietary instant messaging, Voice over IP, e-mail and notification delivery protocols that belong in this category. All of these protocols aim to keep the client-server connection alive for as long as the application is running. When the application has otherwise no traffic to send, specific keepalive messages are sent periodically to ensure that the NAT/FW state in the middle does not expire. The client can use PCP to keep the required mappings at the NAT/FWs and use application keepalives to keep the state on the Application Server/Peer as mentioned in Section 3.4.

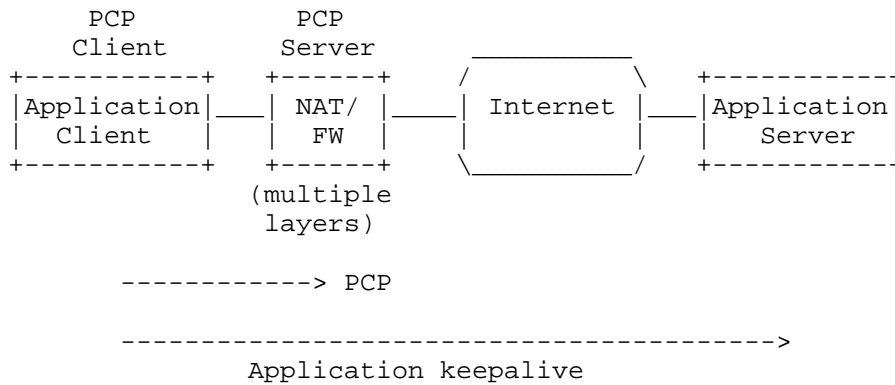


Figure 1: PCP with Client-Server applications

There are also scenarios where the long-term communication association is between two peers, both of whom may reside behind one or more layers of NAT/FW. This is depicted in Figure 2. The initiation of the association may have happened using mechanisms such as Interactive Communications Establishment (ICE), perhaps first triggered by a "signaling" protocol such as SIP or XMPP or WebRTC [I-D.ietf-rtcweb-overview]. Examples of the peer-to-peer protocols include RTP and WebRTC data channel. A number of proprietary VoIP or video call or streaming or file transfer protocols also exist in this category. Typically the communication is based on UDP, but TCP or SCTP may be used. If there is no traffic flowing, the peers have to inject periodic keepalive packets to keep the NAT/FW mappings on both sides of the communication active. Instead of application keepalives, both peers can use PCP to control the mappings on the NAT/FWs to reduce the keepalive frequency as explained in Section 3.4.

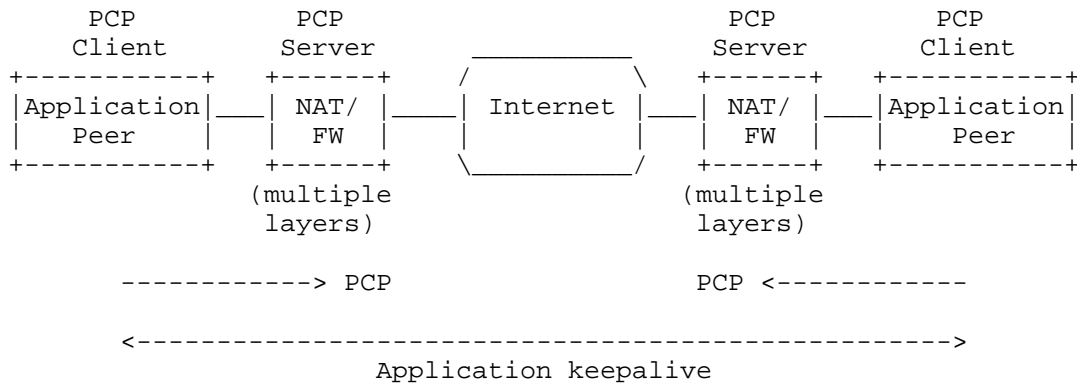


Figure 2: PCP with Peer-to-Peer applications

3.2. NAT Topologies and Detection

Before an application can reduce its keepalive rate, it has to make sure it has all of the NATs and firewalls on its path under control. This means it has to detect the presence of any PCP-unaware NATs and firewalls on its path to the Internet.

3.2.1. PCP based detection

PCP itself is able to detect unexpected NATs between the PCP client and PCP server as depicted in Figure 3. The PCP client includes its own IP address and UDP port within the PCP request. The PCP server compares them to the source IP address and UDP port it sees on the packet. If they differ, there are one or more additional NATs between the PCP client and PCP server, and the server will return an error. Unless the application has some other means (like UPnP) to control these PCP unaware NATs, it has to fall back to its default keepalive mechanism.

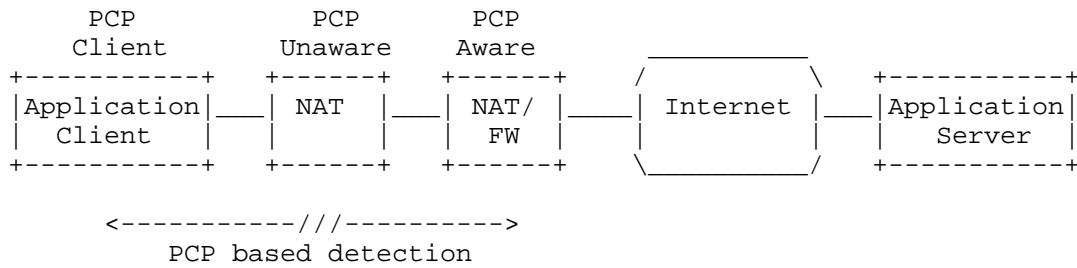


Figure 3: PCP unaware NAT between PCP client and PCP server

3.2.2. Application based detection

Figure 4 shows a topology where one or more PCP unaware NATs are deployed on the exterior of the PCP capable NAT/FWs. To detect this, the application client must have the capability to request from its application server or peer what IP and transport address it sees. If those differ from the IP and transport address given by the PCP aware NAT/FW then the application client can determine that there is at least one PCP unaware NAT on the path. In this case, the application client has to fall back to its default keepalive mechanism.

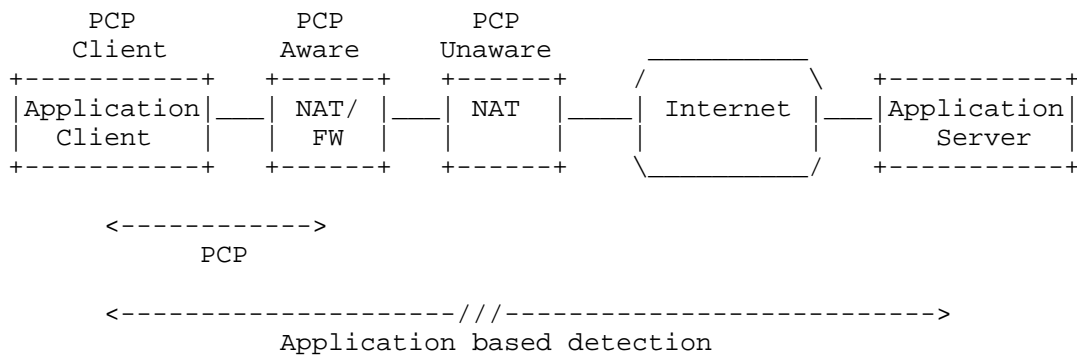


Figure 4: PCP unaware NAT external to the last PCP aware NAT

3.3. Detection of PCP unaware firewalls

PCP and application based detection mechanisms explained in Section 3.2.1 and Section 3.2.2 are based on change in the address and will not detect PCP unaware firewalls. In order to detect a PCP

unaware firewall, the application client sends a Session Traversal Utilities for NAT (STUN) [RFC5389] Binding request to the STUN server. If STUN server supports the STUN extensions defined in [RFC5780] then it returns its alternate IP address and alternate port in OTHER-ADDRESS attribute in the STUN Binding response. The client then uses PCP to send MAP request with FILTER option to PCP server to permit STUN server to reach the client using the STUN servers alternate IP address and alternate port. The client then sends a Binding request to the primary address of the STUN server with the CHANGE-REQUEST attribute set to change-port and change-IP. This will cause the server to send its response from its alternate IP address and alternate port. If the client receives a response then the client is aware that on path firewall devices are PCP aware. If the client does not receive a response then the client is aware that there could be one or more on path PCP unaware firewall devices. The application client will perform the tests separately for each transport protocol. If no response is received, the client will then repeat the test at most three times for connectionless transport protocols.

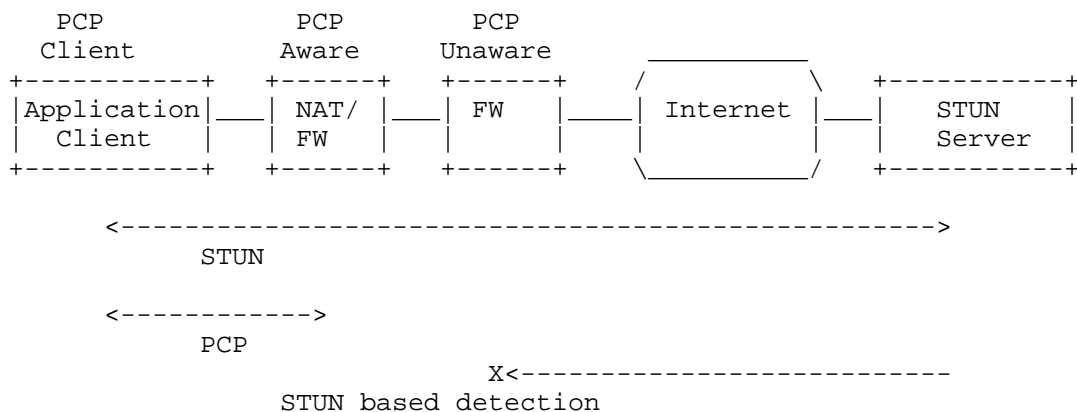


Figure 5: PCP unaware firewall

This procedure can be adopted by other protocols to detect PCP unaware firewalls.

3.4. Keepalive Optimization

If the application determines that all NATs and firewalls on its path to the Internet support PCP, it can start using PCP instead of its default keepalives to maintain the NAT/FW state. It can use PCP PEER

Request with the Requested Lifetime set to an appropriate value. The application may still send some application-specific heartbeat messages end-to-end to refresh state on the application server, which typically requires keepalives far less frequently than NATs /FWs do.

Processing the lifetime value of the PEER Opcode is described in Sections 10.3 and 15 of [RFC6887]. Sending a PEER request with a very short Requested Lifetime can be used to query the lifetime of an existing mapping. PCP recommends that lifetimes of mapping created or lengthened with PEER be longer than the lifetimes of implicitly-created NAT and firewall mappings. Thus PCP can be used to reduce power consumption by making PCP PEER message interval longer than what the application would normally use to the keep the middle box state alive, and strictly shorter than the server state refresh interval.

An example of savings with PCP is described in Appendix B.

4. Keepalive Interval Determination Procedure when PCP unaware Firewall or NAT is detected

If a PCP unaware NAT/firewall is detected, then a client can use the following heuristics method to determine the keepalive interval:

1. The client sends a STUN Binding request to the STUN server. This connection is called the Primary Channel. STUN server will return its alternate IP address and alternate port in OTHER-ADDRESS in the Binding response [RFC5780].
2. The client then sends a STUN Binding request to the STUN server using alternate IP address and alternate port. This connection is called the Secondary Channel.
3. The Client will initially set the default keepalive interval for NAT/FW mappings to 60 seconds (FWa).
4. After FWa seconds the Client will send a Binding request to the STUN server using the Primary Channel with the CHANGE-REQUEST attribute set to change-port and change-IP. This will cause the STUN server to send its response from the Secondary channel.
5. If the client receives response from the server then it will increase the keepalive interval value $FWa = (old\ FWa) + (old\ FWa)/2$. This indicates that NAT/FW mappings are alive.
6. Steps 4 and 5 will be repeated until there is no response from the STUN server. If there is no response from the STUN server

then the client will use the old FWa value as Keepalive interval to refresh FW/NAT mappings.

The above procedure will be done separately for each transport protocol. For connectionless transport protocols such as UDP, if 2 seconds elapse without a response from the STUN server then the client will repeat step 4 at most three times to handle packet loss.

This procedure can be adopted by other protocols to use Primary and Secondary channels, so that the client can determine the keepalive interval to refresh FW/NAT mapping. This procedure only serves as a guideline and if applications already use some other heuristic to determine the keepalive interval, they can continue with the existing logic. For example Teredo determines the Refresh interval using the procedure in "Optional Refresh Interval Determination Procedure" (Section 5.2.7 of [RFC4380]).

Note: The keepalive interval learnt using the above method can be inaccurate if a firewall is configured with an application-specific inactivity timeout.

To improve reliability, applications SHOULD continue to use PCP to lengthen the FW/NAT mappings even if the above mechanism is used to detect PCP unaware NAT/firewall. This ensures that PCP aware FW/NATs do not close old mappings with no packet exchange when there is a resource-scarcity situation.

5. Application-Specific Operation

This section describes how PCP is used with specific application protocols.

5.1. SIP

For connection-less transports the User Agent (UA) sends a STUN Binding request over the SIP flow as described in section 4.4.2 of [RFC5626]. The UA then learns the External IP Address and Port using a PCP PEER request/response. If the XOR-MAPPED-ADDRESS in the STUN Binding response matches the external address and port provided by PCP PEER response then the UA optimizes the keepalive traffic as described in Section 3.4. There is no further need to send STUN Binding requests over the SIP flow to keep the NAT Binding alive.

If the XOR-MAPPED-ADDRESS in the STUN Binding response does not match the external address and port provided by the PCP PEER response then PCP will not be used to keep the NAT bindings alive for the flow that is being used for the SIP traffic. This means that multiple layers of NAT are involved and intermediate NATs are not PCP aware. In this

case the UA will continue to use the technique in section 4.4.2 of [RFC5626].

For connection-oriented transports, the UA sends a STUN Binding request multiplexed with SIP over the TCP connection. STUN multiplexed with other data over a TCP or TLS-over-TCP connection is explained in section 7.2.2 of [RFC5389]. The UA then learns the External IP address and port using a PCP PEER request/response. If the XOR-MAPPED-ADDRESS in the STUN Binding response matches the external address and port provided by the PCP PEER response, then the UA optimizes the keepalive traffic as described in Section 3.4.

If the XOR-MAPPED-ADDRESS in the STUN Binding response does not match the external address and port provided by the PCP PEER response, then PCP will not be used to keep the NAT bindings alive. In this case the UA performs a keepalive check by sending a double-CRLF (the "ping") then waits to receive a single CRLF (the "pong") using the technique in section 4.4.1 of [RFC5626].

5.2. HTTP

Web Applications that require persistent connections use techniques such as HTTP long polling and Websockets for session keep alive as explained in section 3.1 of [I-D.isomaki-rtcweb-mobile]. In such scenarios, after the client establishes a connection with the HTTP server, it can execute server side scripts such as PHP residing on the server to provide the transport address and port of the HTTP client seen at the HTTP server. In addition, the HTTP client also learns the external IP Address and port using a PCP PEER request/response.

If the IP address and port learned from the server matches the external address and port provided by the PCP PEER response then the HTTP client optimizes keepalive traffic as described in Section 3.4.

If the IP address and port do not match, then PCP will not be used to keep the NAT bindings alive for the flow that is being used for the HTTP traffic. This means that there are NATs or HTTP proxies between the PCP server and the HTTP server. The HTTP client will have to resort to use existing techniques for keep alive. Please see Appendix A for an example server side PHP script to obtain the client source IP address.

The HTTP protocol allows intermediaries such as transparent proxies to be involved and there is no way for the client to know that a request/response is relayed through a proxy.

5.3. Media and data channels with ICE

The ICE agent learns the External IP Addresses and Ports using the PCP MAP opcode. If server reflexive candidates learnt using STUN [RFC5389] and external IP addresses learnt using PCP are different then candidates learnt through both STUN and PCP are encoded in the ICE offer and answer. When using the Recommended Formula explained in section 4.1.2.1 of [RFC5245] to compute priority for the candidate learnt through PCP, the ICE agent MUST use a preference value greater than the server reflexive candidate and hence tested before the server reflexive candidate. The recommended type preference value is 105 for candidates discovered using PCP and is explained in section 4.2 of [RFC6544].

The ICE agent, in addition to the ICE connectivity checks, performs the following:

1. The ICE agent checks if the XOR-MAPPED-ADDRESS from the STUN Binding response received as part of ICE connectivity check matches the External IP address and Port provided by PCP MAP response.
2. If the match is successful then PCP will be used to keep the NAT bindings alive. The ICE agent optimizes keepalive traffic by refreshing the mapping via a new PCP MAP request containing information from the earlier PCP response.
3. If the match is not successful then PCP will not be used for keep NAT binding alive. The ICE agent will use the technique in section 4.4 of [RFC6263] to keep NAT bindings alive. This means that multiple layers of NAT are involved and intermediate NATs are not PCP- aware.

Some network operators deploying a PCP Server may allow PEER but not MAP. In such cases the ICE agent learns the external IP address and port using a STUN Binding request/response during ICE connectivity checks. The ICE agent also learns the external IP Address and port using a PCP PEER request/response. If the IP address and port learned from the STUN Binding response matches the external address and port provided by the PCP PEER response then the ICE agent optimizes keepalive traffic as described in Section 3.4.

5.4. Detecting Flow Failure

Using the Rapid Recovery technique in section 14 of [RFC6887] upon receiving a PCP ANNOUNCE from a PCP server, a PCP client becomes aware that the PCP server has rebooted or lost its mapping state. The PCP client issues new PCP requests to recreate any lost mapping

state and thus reconstructs lost mappings fast enough that existing media, HTTP and SIP flows do not break. If the NAT state cannot be recovered the endpoint will find the new external address and port as part of the Rapid Recovery technique in PCP itself and reestablish a connection with the peer.

5.5. Firewalls

PCP allows applications to communicate with firewall devices with PCP functionality to create mappings for incoming connections. In such cases PCP can be used by the endpoint to create an explicit mapping on firewall in order to permit inbound traffic. The endpoint can further use PCP to send keepalives to keep the firewall mappings alive.

5.5.1. IPv6 Network with Firewalls

For scenarios where the client uses the ICE Lite implementation explained in section 2.7 of [RFC5245], the ICE Lite endpoint will not generate its own ICE connectivity checks, by definition. As part of the call setup, the ICE Lite endpoint would gather its host candidates and relayed candidate from a TURN server and send the candidates in the offer to the peer endpoint. On receiving the answer from the peer endpoint, the ICE Lite endpoint sends a PCP MAP request with FILTER opcode to create a dynamic mapping in the firewall to permit ICE connectivity checks and subsequent media traffic from the remote peer. This way, the ICE Lite endpoint and its network are protected from unsolicited incoming UDP traffic, and can still operate using ICE Lite (rather than full ICE).

5.5.2. Mobile Network with Firewalls

Some mobile networks are also making use of a firewall to protect their customers from various attacks like downloading malicious content. The firewall is usually configured to block all unknown inbound connections as explained in section 2.1 of [I-D.chen-pcp-mobile-deployment]. As described in Section 3.4, in such cases, PCP can be used by mobile devices to create an explicit mapping on the firewall to permit inbound traffic and optimize the keepalive traffic. This would result in saving of radio and power consumption of the mobile device while protecting it from attacks.

6. IANA Considerations

This document has no actions for IANA.

7. Security Considerations

The security considerations in [RFC5245] and [RFC6887] apply to this use.

8. Acknowledgements

Authors would like to thank Dave Thaler, Basavaraj Patil, Anca Zamfir, Reinaldo Penno, Suresh Kumar, Dilipan Janarthanan and Mohamed Boucadair for their valuable inputs.

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.
- [RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.
- [RFC5626] Jennings, C., Mahy, R., and F. Audet, "Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)", RFC 5626, October 2009.
- [RFC5780] MacDonald, D. and B. Lowekamp, "NAT Behavior Discovery Using Session Traversal Utilities for NAT (STUN)", RFC 5780, May 2010.
- [RFC6263] Marjou, X. and A. Sollaud, "Application Mechanism for Keeping Alive the NAT Mappings Associated with RTP / RTP Control Protocol (RTCP) Flows", RFC 6263, June 2011.
- [RFC6544] Rosenberg, J., Keranen, A., Lowekamp, B., and A. Roach, "TCP Candidates with Interactive Connectivity Establishment (ICE)", RFC 6544, March 2012.
- [RFC6887] Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, April 2013.

9.2. Informative References

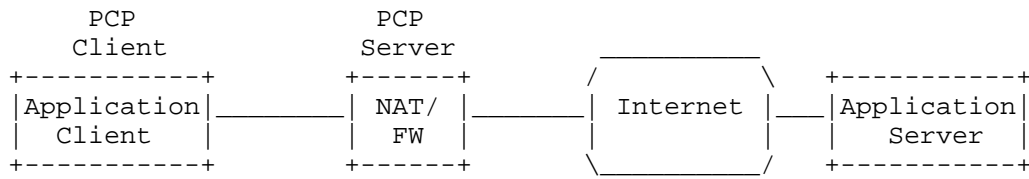
- [I-D.chen-pcp-mobile-deployment]
Chen, G., Cao, Z., Boucadair, M., Ales, V., and L. Thiebaut, "Analysis of Port Control Protocol in Mobile Network", draft-chen-pcp-mobile-deployment-04 (work in progress), July 2013.
- [I-D.ietf-rtcweb-overview]
Alvestrand, H., "Overview: Real Time Protocols for Browser-based Applications", draft-ietf-rtcweb-overview-13 (work in progress), November 2014.
- [I-D.ietf-v6ops-mobile-device-profile]
Binet, D., Boucadair, M., Ales, V., Chen, G., Heatley, N., Chandler, R., Michaud, D., Lopez, D., and W. Haeffner, "An Internet Protocol Version 6 (IPv6) Profile for 3GPP Mobile Devices", draft-ietf-v6ops-mobile-device-profile-21 (work in progress), March 2015.
- [I-D.isomaki-rtcweb-mobile]
Isomaki, M., "RTCweb Considerations for Mobile Devices", draft-isomaki-rtcweb-mobile-00 (work in progress), July 2012.
- [RFC2177] Leiba, B., "IMAP4 IDLE command", RFC 2177, June 1997.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
- [RFC3921] Saint-Andre, P., Ed., "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence", RFC 3921, October 2004.
- [RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, February 2006.
- [RFC6455] Fette, I. and A. Melnikov, "The WebSocket Protocol", RFC 6455, December 2011.

Appendix A. Example PHP script

```
<html>
Connected to <?PHP echo gethostname(); ?> on port <?PHP echo
getenv(SERVER_PORT)?> on <?PHP echo date("d-M-Y H:i:s");?>
Pacific Time
<p>
Your IP address is: <?PHP echo getenv(REMOTE_ADDR); ?>,
port <?PHP echo getenv(REMOTE_PORT); ?>
</p>;
</html>
```

Appendix B. Savings with PCP

The following example illustrates the savings in keepalive messages with PCP.



With Application Heartbeat (without PCP):

```

<-----//----->
  Application heartbeat (Max Interval = 30 seconds)
<-----//----->
  Application heartbeat (Max Interval = 30 seconds)
<-----//----->
  Application heartbeat (Max Interval = 30 seconds)
<-----//----->
  Application heartbeat (Max Interval = 30 seconds)
  ....
  ....
  ....
  ....
  
```

With PCP:

```

<----->
  PCP PEER request
  (Max Lifetime = 3600 seconds)
  ....
  ....
<----->
  PCP PEER request
  (Max Lifetime = 3600 seconds)
  
```

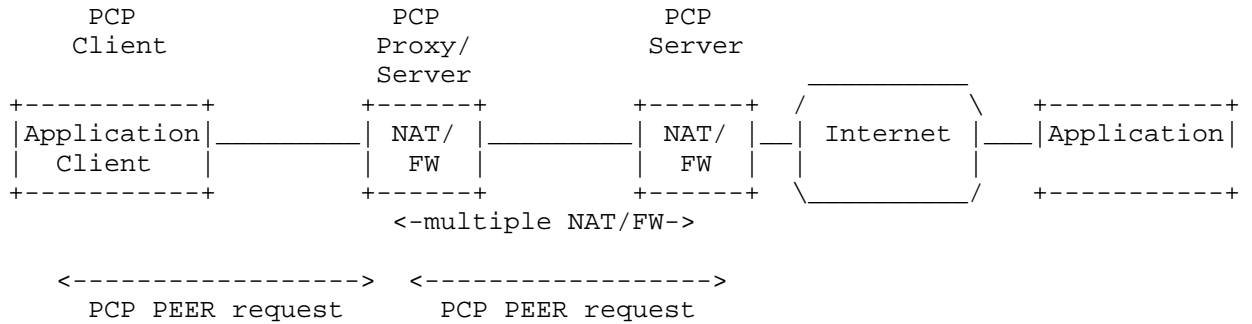
Figure 6: Savings with PCP

In the example above, let's suppose normally an application would need to send a heartbeat every 30s to keep mappings active on the NAT/firewall device. In 24 hours, in the absence of PCP, the number of packets sent by the application to keep those mappings active would be $(86400/30) = 2880$ packets.

If the same application uses PCP PEER to create a mapping, with a lifetime of 3600 seconds, on a PCP controlled NAT/firewall device, the number of packets sent by the application to keep those mappings active would be $(86400/3600) = 24$ packets.

With the above assumptions, using PCP saves 99.16% of keepalive traffic. As the number of applications running on a host increase,

savings in cost of sending application heartbeats are significant with the use of PCP.



If there are multiple PCP-aware NAT/firewall devices on a client's path to the internet, e.g., PCP servers at a home gateway and also at a CGN, the savings with PCP are the same. The PCP server at the home gateway is a PCP proxy that can create associated mappings on the PCP server at the CGN. The client will only have to communicate with the PCP proxy, and receives a single mapping lifetime that needs to be refreshed.

Authors' Addresses

Tirumaleswar Reddy
 Cisco Systems, Inc.
 Cessna Business Park, Varthur Hobli
 Sarjapur Marathalli Outer Ring Road
 Bangalore, Karnataka 560103
 India

Email: tireddy@cisco.com

Prashanth Patil
 Cisco Systems, Inc
 Bangalore
 India

Email: praspati@cisco.com

Markus Isomaki
Nokia
Keilalahdentie 2-4
FI-02150 Espoo
Finland

Email: markus.isomaki@nokia.com

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

Email: dwing@cisco.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: April 30, 2015

A. Ripke
T. Dietz
J. Quittek
NEC
R. da Silva
Telefonica I+D
October 27, 2014

PCP Third Party ID Option
draft-ripke-pcp-tunnel-id-option-02

Abstract

This document describes a new Port Control Protocol (PCP) option called `THIRD_PARTY_ID`. It serves for identifying a Third Party in addition to the means that PCP's `THIRD_PARTY` option already provides for that purpose.

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 30, 2015.

Copyright Notice

Copyright (c) 2014 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

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
3. Target Scenarios	3
3.1. Carrier-hosted UPnP IGD-PCP IWF	5
3.2. Carrier Web Portal	6
3.3. Other Use Cases	6
4. Format	6
5. Behavior	8
5.1. Generating a Request	8
5.2. Processing a Request	8
5.3. Processing a Response	8
6. Alternative	8
7. IANA Considerations	8
8. Security Considerations	9
9. References	9
9.1. Normative References	9
9.2. Informative References	9
Authors' Addresses	10

1. Introduction

The IETF has specified the Port Control Protocol (PCP) ([RFC6887]) to control how packets are translated and forwarded by a PCP-controlled device such as a network address translator (NAT) or firewall.

This draft focuses on the application of PCP's `THIRD_PARTY` option that is used when the PCP client sends requests that concern other internal hosts than the host of the PCP client. This is, for example, the case if port mapping requests for a carrier grade NAT (CGN) are not sent from PCP clients at the subscribers, but from a PCP Interworking Function which requests port mappings.

The issue addressed by the `THIRD_PARTY_ID` option is that there are CGN deployments that do not distinguish internal hosts by their IP address only, but use further identifiers for unique subscriber identification. This is, for example, the case if a CGN supports overlapping private IP address spaces according to [RFC1918] for internal hosts of different subscribers. Then different internal hosts are identified and mapped at the CGN by their IP address and an additional ID, for example, the ID of a tunnel between the CGN and the subscriber. In such cases, the IP address contained in the `THIRD_PARTY` option is not sufficient. An additional identifier needs

to be carried by the PCP protocol in order to uniquely identify the internal host. The THIRD_PARTY_ID option serves this purpose.

The THIRD_PARTY_ID option is defined for use in combination with the THIRD_PARTY option for the PCP opcodes MAP and PEER.

We renamed the option name from TUNNEL_ID to THIRD_PARTY_ID to reflect the fact that this identifier is an extended THIRD_PARTY option for general applicability.

2. Terminology

The terminology defined in the specification of PCP [RFC6887] applies.

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 RFC 2119 [RFC2119].

3. Target Scenarios

This section describes two scenarios that illustrate the use of the THIRD_PARTY_ID option:

1. a UPnP IGD-PCP IWF (Universal Plug and Play Internet Gateway Device - Port Control Protocol Interworking Function),
2. a carrier web portal for port mapping.

Both scenarios are variants of the same basic scenario shown in Figure 1. It has a carrier operating a CGN and a Port Control Protocol Interworking Function (PCP IWF) for subscribers to request port mappings at the CGN. The PCP IWF communicates with the CGN using PCP. For this purpose the PCP IWF contains a PCP client and the CGN is co-located with a PCP server. The way subscribers interact with the PCP IWF for requesting port mapping for their internal hosts is not specified in this basic scenario, but more elaborated in the specific scenarios below.

information in its internal mapping tables for translating packet addresses and for forwarding packets to subscriber-specific tunnels.

How the carrier PCP IWF is managing port mappings, such as, for example, automatically extending the lifetime of a mapping, is beyond the scope of this document.

3.1. Carrier-hosted UPnP IGD-PCP IWF

This scenario further elaborates the basic one above by choosing UPnP as communication protocol between subscriber and the carrier's PCP IWF. Then obviously, the PCP IWF is realized as an UPnP IGD-PCP IWF as specified in [RFC6970].

As shown in Figure 2 it is assumed here that the UPnP IGD-PCP IWF is not embedded in the subscriber premises router, but offered as a service to the subscriber. Further, it is assumed that the UPnP IGD-PCP IWF is not providing NAT functionality.

This requires that the subscriber has a UPnP connection to the UPnP IGD-PCP IWF, which can, for example, be provided via (one of the) tunnel(s) connecting the subscriber's network to the CGN. This connection can then be used by hosts in the subscriber's network to request port mappings at the CGN using UPnP as specified in [RFC6970].

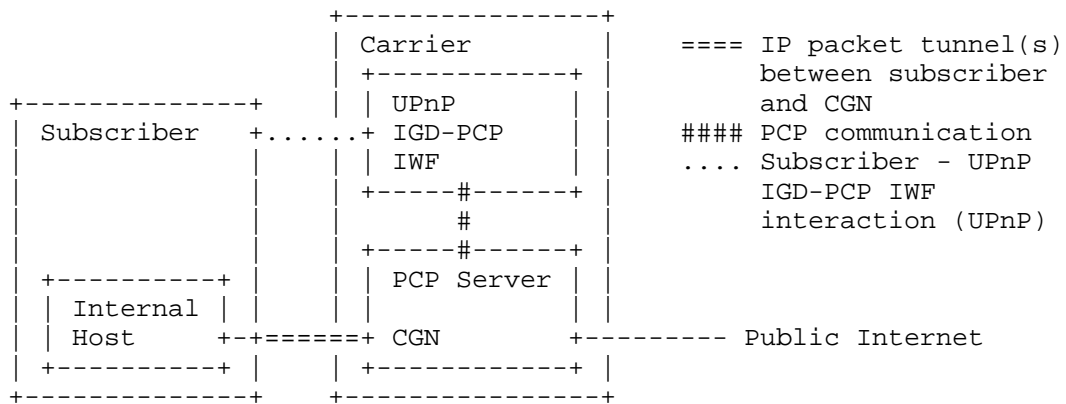


Figure 2: UPnP IGD-PCP IWF

A potential extension to [RFC6970] regarding an additional state variable for the THIRD_PARTY_ID and regarding an additional error code for a mismatched THIRD_PARTY_ID and its processing might be a logical next step. However, this is not in the scope of this document.

5. Behavior

The following sections describe the operations of a PCP client and a PCP server when generating the request and processing the request and response.

5.1. Generating a Request

In addition to generating a PCP request that is described in [RFC6887] the following has to be applied. The THIRD_PARTY_ID option can be used together either with a PCP MAP or PEER opcode. It MUST be used in combination with the THIRD_PARTY option which provides an IP address and port entered by the subscriber. The THIRD_PARTY_ID option holds an identifier to allow the CGN to uniquely identify the internal host (specified in the THIRD_PARTY option) for which the port mapping is to be established or modified. If the identifier is shorter than 128 bits then the THIRD_PARTY_ID option field is to be filled up with leading zeros up to 128 bits.

5.2. Processing a Request

The THIRD_PARTY_ID option is in the mandatory-to-process range and if the PCP server does not support this option it MUST return an UNSUPP_OPTION response. If the provided THIRD_PARTY_ID is unknown/unavailable the PCP server MUST return a THIRD_PARTY_ID_UNKNOWN response.

5.3. Processing a Response

If the PCP client receives a THIRD_PARTY_ID_UNKNOWN response back for its previous request it SHOULD report an error message. To where to report an error message is implementation dependent.

6. Alternative

An alternative to identify a tunnel affiliation in the given scenario could be using the DESCRIPTION ([RFC7220]) option to carry a tunnel ID option. The DESCRIPTION option is to allow a text description to be attached to a port mapping. But using the DESCRIPTION option for a tunnel ID might not be appropriate because it specifies using UTF-8 and another requirement is that the description text must not be null terminated, which cannot always be met.

7. IANA Considerations

The following PCP Option Code is to be allocated in the mandatory-to-process range:

THIRD_PARTY_ID

[NOTE for IANA: Please allocate a PCP Option Code at <http://www.iana.org/assignments/pcp-parameters/pcp-parameters.xml#option-rules>]

The following PCP Result Code is to be allocated:

THIRD_PARTY_ID_UNKNOWN

[NOTE for IANA: Please allocate a PCP Result Code at <http://www.iana.org/assignments/pcp-parameters/pcp-parameters.xml#result-codes>]

8. Security Considerations

As this option is related to the use of the THIRD_PARTY option the corresponding security considerations apply. Especially, the network on which the PCP messages are sent must be fully trusted.

9. References

9.1. Normative References

- [RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6887] Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, April 2013.

9.2. Informative References

- [I-D.boucadair-pcp-deployment-cases] Boucadair, M., "Port Control Protocol (PCP) Deployment Models", draft-boucadair-pcp-deployment-cases-03 (work in progress), July 2014.
- [I-D.boucadair-pcp-sfc-classifier-control] Boucadair, M., "PCP as a Traffic Classifier Control Protocol", draft-boucadair-pcp-sfc-classifier-control-01 (work in progress), October 2014.

[RFC6970] Boucadair, M., Penno, R., and D. Wing, "Universal Plug and Play (UPnP) Internet Gateway Device - Port Control Protocol Interworking Function (IGD-PCP IWF)", RFC 6970, July 2013.

[RFC7220] Boucadair, M., Penno, R., and D. Wing, "Description Option for the Port Control Protocol (PCP)", RFC 7220, May 2014.

Authors' Addresses

Andreas Ripke
NEC
Heidelberg
Germany

Email: ripke@neclab.eu

Thomas Dietz
NEC
Heidelberg
Germany

Email: dietz@neclab.eu

Juergen Quittek
NEC
Heidelberg
Germany

Email: quittek@neclab.eu

Rafael Lopez da Silva
Telefonica I+D
Madrid
Spain

Email: ralds@tid.es

Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 4, 2016

S. Vinapamula
Juniper Networks
S. Sivakumar
Cisco Systems
M. Boucadair
Orange
T. Reddy
Cisco
November 1, 2015

Application-Initiated Flow High Availability Awareness through Port
Control Protocol (PCP)
draft-vinapamula-flow-ha-14

Abstract

This document specifies a mechanism for a host to signal via Port Control Protocol (PCP) which connections should be protected against network failures. These connections will be elected to be subject to high availability mechanisms enabled at the network side.

This approach assumes that applications/users have more visibility about sensitive connections rather than any heuristic that can be enabled at the network side to guess which connections should be check-pointed.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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 May 4, 2016.

Copyright Notice

Copyright (c) 2015 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 the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Note	3
2. Issues with the Existing Implementations	3
3. CHECKPOINT-REQUIRED PCP Option	4
3.1. Format	4
3.2. Operation	5
4. Sample Use cases	7
5. Security Considerations	8
6. IANA Considerations	9
7. References	9
7.1. Normative references	9
7.2. Informative References	9
Appendix A. Appendix	11
Acknowledgments	11
Authors' Addresses	11

1. Introduction

The risk of Internet service disruption is critical in service providers and enterprise networking environments. Such a risk is often mitigated with the introduction of active/backup systems. Such designs not only contribute to minimize the risk of service disruption, but also facilitate maintenance operations (e.g., hitless H/W or S/W upgrades).

In addition, the nature of some connections leads to the establishment and the maintenance of connection-specific states by some of the network functions invoked when the connection is established. During active/backup failover in case of a network

failure, the said states need to be check-pointed by the backup system. Additional issues are further discussed in Section 2.

Heuristics based on the protocol, mapping lifetime, etc., are used in the network to elect which connections need to be check-pointed (e.g., by means of high availability techniques). This document advocates for an application-initiated approach that would allow applications/users to signal to the network which of their connections are critical.

This document specifies how PCP [RFC6887] can be extended to signal which connection should be check-pointed for high availability (Section 3). A set of use cases are provided for illustration purposes in Section 4. This document does not make any assumption on the PCP-controlled device that will process the PCP-formatted signaling information from PCP clients. These devices are likely to be flow-aware.

The approach in this document is aligned with the networking trends advocating for open network APIs to interact with applications/services (e.g., [RFC7149]). Policy-decision making process at the network side will be enriched with information signaled by application using PCP for instance.

1.1. Note

The CHECKPOINT-REQUIRED PCP option (Section 3) is defined in the Specification Required range (see Section 6). In order to be assigned a code point in that range, a permanent publication is required as per Section 4.1 of [RFC5226]. Publication of an RFC is an ideal means of achieving this requirement and also to ease interoperability.

Note, this work was presented to the Port Control Protocol (pcp) WG but there was no consensus to define this option in the "Standards Action" range despite positive feedback was received from the working group. Technical comments that were received during pcp meetings and those received on the mailing list were addressed.

2. Issues with the Existing Implementations

Regardless of the selected technology or design like HA-based designs, reliably securing connections is expensive in terms of memory, CPU and other resources. Also check-pointing may not be required for all connections as all connections may not be critical. But, this leaves a challenge to identify what connections to check-point.

Typically, long-lived connections are identified and, only the states of such connections are check-pointed.

Typically, this is addressed by identifying long lived connections and check-pointing state of only those connections that lived long enough, to the backup for service continuity.

However, check-pointing long lived connections raises the following issues:

1. It is hard for a network to identify/guess which connection is (business) critical. This characterization is often customer-specific: a flow can be sensitive for a User#1 while it is not for another User#2. Furthermore, this characterization can vary over time: a flow can be sensitive during hour X, while it is not be during other times.
2. Heuristics are not deterministic.
3. A potentially long-lived connection may experience disruption upon failure of the active system, but before it is check-pointed.
4. A connection may not be long lived but critical Voice over IP (VoIP) conversations.
5. Likewise, not all long-lived connections are deemed critical: for example, connections that pertain to free Internet services are usually considered not critical compared to the equivalent connections for paid services. Only the latter need to be check-pointed.

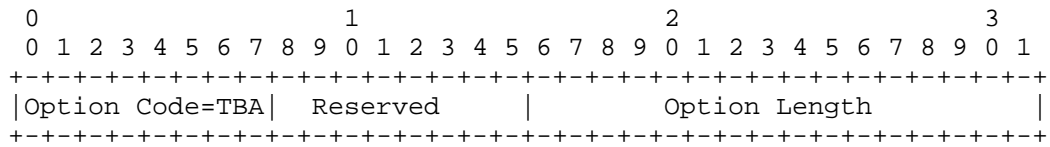
3. CHECKPOINT-REQUIRED PCP Option

3.1. Format

The solution is based on the assumption that an application or user is the best judge to decide which of its connections are critical.

An application or user may explicitly identify the connections that need to be check-pointed by means of a PCP client, using the CHECKPOINT_REQUIRED option as described in Figure 1.

The entry to be backed up is indicated by the content of a MAP or PEER message.



Option Name: CHECKPOINT_REQUIRED
 Number: <TBA>
 Purpose: Indicate if an entry needs to be check-pointed.
 Valid for Opcodes: MAP, PEER
 Length: 0.
 May appear in: request, response.
 Maximum occurrences: 1.

Figure 1: CHECKPOINT_REQUIRED PCP Option

The description of the fields is as follows:

- o Option Code: To be assigned by IANA (see Section 6).
- o Reserved: This field is initialized as specified in Section 7.3 of [RFC6887].
- o Option Length: 0. This means no data is included in the option.

An application or user can take advantage of this PCP option to explicitly indicate which of the connections need to be check-pointed and should not be disrupted. The processing of this option by the PCP server will then yield the check-pointing of the corresponding states by the relevant devices or functions dynamically controlled by the PCP server.

Communication between application/user and PCP client is implementation-specific.

3.2. Operation

Support of the CHECKPOINT_REQUIRED option by PCP servers and PCP clients is optional. This option (Code TBA; see Figure 1) may be included in a PCP MAP/PEER request to indicate a connection is to be protected against network failures.

There is a risk that every PCP client may wish to check-point every connection, which can potentially load the system. Administration SHOULD restrict the number of connections that can be elected to be backed up and the rate of check-pointing on per network attachment point (e.g., CPE, host). To that aim, the PCP server should unambiguously identify the network attachment point a PCP client

belongs to. For example, the PCP server may rely on the PCP identity [RFC7652], the assigned prefix to a CPE/host, the subscriber-mask [I-D.vinapamula-softwire-dslite-prefix-binding], or other identification means.

The PCP client includes a CHECKPOINT_REQUIRED option in a MAP or PEER request to signal that the corresponding mapping is to be protected.

If the PCP client does not receive a CHECKPOINT_REQUIRED option in response to a PCP request that enclosed the CHECKPOINT_REQUIRED option, this means that either the PCP server does not support the option, or the PCP server is configured to ignore the option or the PCP server cannot satisfy the request expressed in this option (e.g., because of a lack of resources).

If the CHECKPOINT_REQUIRED option is not included in the PCP client request, the PCP server MUST NOT include the CHECKPOINT_REQUIRED option in the associated response.

When the PCP server receives a CHECKPOINT_REQUIRED option, the PCP server checks if it can honor this request depending on whether resources are available for check-pointing. If there are no resources available for check-pointing, but there are resources available to honor the MAP/PEER request, a response is sent back to the PCP client without including the CHECKPOINT_REQUIRED option (i.e., the request is processed as any MAP/PEER request that does not convey a CHECKPOINT_REQUIRED option). If check-pointing resources are still available and the quota for this PCP client is not reached, the PCP server tags the corresponding entry as eligible to HA mechanism and sends back the CHECKPOINT_REQUIRED option in the positive answer to the PCP client.

To update the check-pointing behavior of a mapping maintained by the PCP server, the PCP client generates a PCP MAP/PEER renewal request that includes a CHECKPOINT_REQUIRED option to indicate this mapping has to be check-pointed or without including a CHECKPOINT_REQUIRED option to indicate this mapping does not need be check-pointed anymore. Upon receipt of the PCP request, the PCP server proceeds with the same operations to validate a MAP/PEER request updating an existing mapping. If validation checks are passed, the PCP server updates the check-point flag associated with that mapping accordingly (i.e., it is set if a CHECKPOINT_REQUIRED option was included in the update request or it is cleared if no CHECKPOINT_REQUIRED option was included) , and the PCP server returns the response to the PCP client accordingly.

What information to check-point and how to check-point is out of scope of this document, and is left for implementations. Also,

interest to indicate check-pointing by users/applications in a PCP request, may be automatic, semi-automatic, or human intervened. This behavior is also left for application implementations. For managed CPEs, a service provider may influence what connections to be check-pointed.

It is RECOMMENDED to check-point state on backup for honored requests before a response is sent to the PCP client.

4. Sample Use cases

Below are provided some examples for illustration purposes:

Example 1: Consider a streaming service such as live TV broadcasting, or any other media streaming, that supports check-pointing signalling functionality. Suppose, this application is installed in three hosts A, B and C. For A it is critical and doesn't want interruption while for B it is not. While for C, only some programs are of interest. At the time of installing this application's software, corresponding preferences can be provisioned. When the application starts streaming:

- * All the flows associated with the streaming application are critical for A. Limiting the number of flows to be backed up will ensure that host doesn't exceed the user's limit.
- * In case of B, none of these flows are critical for check-pointing. CHECKPOINT_REQUIRED option is not included in the PCP requests.
- * In case of C, the user is invited to interact with the application by the means of a configuration option that is provided to dynamically select which streaming to check-point, based on the user's interest.

Example 2: Consider a streaming service offered by a provider. Suppose, three levels of subscriptions are offered by that provider: e.g., gold, silver, bronze. To guarantee a certain level of quality of service for each subscription, policies are configured such that:

- * All flows associated with a gold subscription should be check-pointed.
- * Only some flows associated with a silver subscription are check-pointed.

- * None of the flows associated with a bronze subscription are check-pointed.

When a user invokes the streaming service, he/she may fall into one of those buckets, and according to the configured policy, his/her associated streaming flows are automatically check-pointed. Login credentials can be used as a trigger to determine the subscription level (and therefore the associated check-pointing behavior).

Example 3: Consider a VoIP application that is able to request its flows to be check-pointed. No matter what is configured by the user, some calls such as emergency calls should be check-pointed. The application has to identify such calls.

Example 4: In the context of an enterprise network, applications are customized by the administrator. Instructions whether a CHECKPOINT_REQUIRED option is to be included is determined by the administrator. Only the subset of applications identified by the administrator will make use of this option in conformance with the enterprise network management policies. Any mis-behavior can be considered as an abuse.

In order to avoid that every application includes a CHECKPOINT_REQUIRED option in its PCP requests, the following items are assumed:

- o Applications may be delivered with some default settings for check-pointing, and these settings should be programmable by end user.
- o Exposing and enforcing these settings is application specific.
- o End user may customize these settings on need basis based on his preferences.

5. Security Considerations

PCP-related security considerations are discussed in [RFC6887].

CHECKPOINT_REQUIRED option can be used by an attacker to identify critical flows, which is sensitive from a privacy standpoint. Also, an attacker can cause critical flows to not be check-pointed by stripping the CHECKPOINT_REQUIRED option or by consuming the quota by adding the option to other flows.

These two issues can be mitigated if the network on which the PCP messages are to be sent is fully trusted. Means to defend against

attackers who can intercept packets between the PCP server and the PCP client should be enabled. In some deployments, access control lists (ACLs) can be installed on the PCP client, PCP server, and the network between them, so those ACLs allow only communications between trusted PCP elements. If the networking environment between the PCP client and the PCP server is not secure, PCP authentication [RFC7652] MUST be enabled.

A network device can always override the end-user signalling, i.e., what is signaled by the PCP client, if the instructions are conflicting with the network policies.

6. IANA Considerations

The following PCP Option Code is to be allocated in the "Specification Required" range (192-223; optional to process range) (the registry is maintained in [http://www.iana.org/ assignments/pcp-parameters](http://www.iana.org/assignments/pcp-parameters)):

CHECKPOINT_REQUIRED set to TBA (see Section 3.1)

7. References

7.1. 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>>.
- [RFC7652] Cullen, M., Hartman, S., Zhang, D., and T. Reddy, "Port Control Protocol (PCP) Authentication Mechanism", RFC 7652, DOI 10.17487/RFC7652, September 2015, <<http://www.rfc-editor.org/info/rfc7652>>.

7.2. Informative References

- [I-D.vinapamula-softwire-dslite-prefix-binding] Vinapamula, S. and M. Boucadair, "Recommendations for Prefix Binding in the Softwire DS-Lite Context", draft-vinapamula-softwire-dslite-prefix-binding-12 (work in progress), October 2015.

- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <<http://www.rfc-editor.org/info/rfc5226>>.
- [RFC7149] Boucadair, M. and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment", RFC 7149, DOI 10.17487/RFC7149, March 2014, <<http://www.rfc-editor.org/info/rfc7149>>.

Appendix A. Appendix

It was tempting to include additional fields in the option but this would lead to a more complex design that is not justified, e.g.,:

- o Define a dedicated field to indicate a priority level. This priority is intended to be used by the PCP server as a hint when processing a request with a CHECKPOINT_REQUIRED option. Nevertheless, an applications may systematically choose to set the priority level to the highest value so that it increases its chance to be serviced!
- o Return a more granular failure error code to the requesting PCP client. Nevertheless this would require extra processing at both the PCP client and server sides for handling the various error codes without any guarantee for the PCP client to have its mappings check-pointed.

Acknowledgments

Thanks to Reinaldo Penno, Stuart Cheshire, Dave Thaler, Prashanth Patil, and Christian Jacquenet for their comments.

Authors' Addresses

Suresh Vinapamula
Juniper Networks
1194 North Mathilda Avenue
Sunnyvale, CA 94089
USA

Phone: +1 408 936 5441
EMail: sureshk@juniper.net

Senthil Sivakumar
Cisco Systems
7100-8 Kit Creek Road
Research Triangle Park, NC 27760
USA

Phone: +1 919 392 5158
EMail: ssenthil@cisco.com

Mohamed Boucadair
Orange
Rennes 35000
France

EMail: mohamed.boucadair@orange.com

Tirumaleswar Reddy
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marathalli Outer Ring Road
Bangalore, Karnataka 560103
India

EMail: tireddy@cisco.com