PCP
Internet-Draft
Intended status: Standards Track
Expires: June 18, 2015

T. Reddy
P. Patil
Cisco
M. Boucadair
France Telecom
December 15, 2014

PCP Firewall Control in Managed Networks
draft-reddy-pcp-sdn-firewall-00

Abstract

In the context of ongoing efforts to add more automation and promote means to dynamically interact with network resources, e.g., SDN-labeled efforts, various proposals are made to accommodate the needs of Network Providers to program the network with flow information. This document describes a means for an SDN controller to install firewall rules using the Port Control Protocol (PCP).

Table of Contents

1. Introduction

All modern firewalls allow an administrator to change the policies in the firewall devices, although the ease of administration for making those changes, and the granularity of the policies, vary widely between firewalls and vendors. With the advent of Software-Defined Networking (SDN), which is a new approach for network programmability, it becomes important to have a means to program these firewalls in a generic fashion. Network programmability in the context of a firewall refers to the capacity to initialize, control, change, and manage firewall policies dynamically via open interfaces as opposed to relying on closed-box solutions and their associated proprietary interfaces.

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. PCP can be leveraged to program firewalls, for example, from an SDN controller using standardized mechanisms.

Existing PCP methods, such as PCP THIRD PARTY OPTION, can be used to install firewall rules, but current PCP methods only allow to create firewall rules on a per-user basis. This document proposes a new PCP OPCODE to accommodate generic firewall based on standard traffic

selectors as described in [RFC6088]. Note, PCP MAP/PEER OpCodes can be used to achieve basic firewall control functionalities, but advanced functionalities are not supported in [RFC6887]. Concretely,[I-D.boucadair-pcp-sfc-classifier-control] identifies some missing PCP features to address the firewall control needs: (1) Extended THIRD_PARTY option to include a L2 identifier (e.g., MAC address), an opaque subscriber-identifier, an IMSI, etc.; (2) Extended FILTER to include a remote range of ports; and (3) DSCP-based filtering. The encoding in Section 3 and the support of the THIRD_PARTY_ID ([I-D.ripke-pcp-tunnel-id-option]) covers most of these functionalities.

PCP extensions in this document can be used in non-SDN contexts such as managed networks. The following use-cases describe the need for SDN controlled firewalls.

1.1. Cloud conferencing server

In the field of real-time conferencing there is a transformation towards cloud-based, virtualized and software based conferencing server implementations. The trend of using virtualized cloud-based services (e.g., conferencing) has a number of positive effects on flexibility, CAPEX, ease of use, etc. One enabling factor for cloud conferencing server is the increased capabilities of the end-points that allow them to decode and process multiple simultaneously received media streams. This in turn has made the central conferring media nodes to switch from mixing or composing media in the decoded domain to instead perform the much less heavy-weight operation of selection, switching and forwarding of media streams, at least for video. Cloud conferencing server typically supports Interactive Connectivity Establishment (ICE) [RFC5245] or at a minimum supports the ICE LITE functionality as described in section 2.7 of [RFC5245]. A cloud conferencing server can terminate ICE and thus have two ICE contexts with either endpoints. The reason for ICE termination is the need for cloud conferencing server to be in the media path. Cloud conferencing server advertises support for ICE in offer/answer and includes its candidates of different types for each component of each media stream.

Enterprise leveraging cloud conferencing server may have a restricted firewall policy to only permit UDP traffic to cloud conferencing provided candidate addresses. The problem is that these candidate addresses could keep changing causing the firewall policy to be frequently modified with human intervention. This problem can be solved by the cloud conferencing server communicating its media candidate addresses to the SDN controller in the enterprise network through a cloud connector and the SDN controller in-turn configures

enterprise firewalls using PCP to permit UDP traffic to the cloud conferencing provided candidate addresses.

1.2. TURN server

Traversal Using Relay NAT (TURN) [RFC5766] is a protocol that is often used to improve the connectivity of Peer-to-Peer (P2P) applications. TURN allows a connection to be established when one or both sides is incapable of a direct P2P connection. The TURN server is a building block to support interactive, real-time communication using audio, video, collaboration, games, etc., between two peer web browsers using the Web Real-Time Communication (WebRTC) framework explained in [I-D.ietf-rtcweb-overview]. A TURN server could be provided by an enterprise network, an access network, an application service provider or a third party provider.

Enterprise that has business agreement with an application or third party provider hosting TURN servers may have a firewall policy to only permit UDP traffic to the external TURN servers provided by the application or third party provider. But the problem is that these TURN addresses could keep changing resulting in the firewall rules to be frequently modified with human intervention. This problem can be solved by the provider hosting the TURN servers to communicate the TURN server IP addresses to the SDN controller deployed in the enterprise network through a cloud connector and the SDN controller in-turn configures enterprise firewalls using PCP to permit UDP traffic to the TURN servers.

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].

3. TSELECT OPCODE

3.1. TSELECT OpCode Packet Format

Figure 1 shows the format of the TSELECT Opcode-specific information.

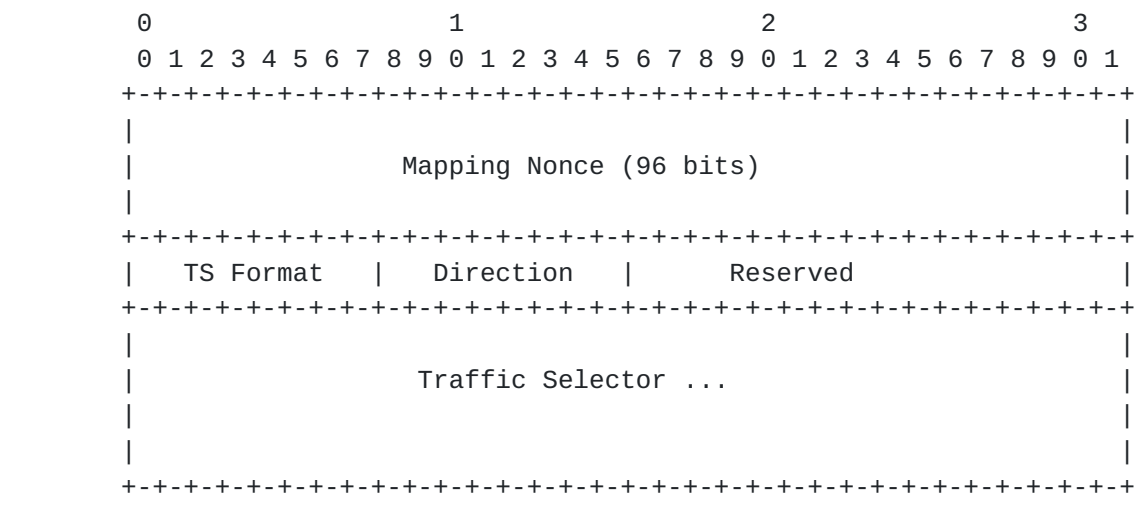

Figure 1: TSELECT Opcode Request

The fields are described below:

Requested/Assigned lifetime (in common header): Requested lifetime of this firewall control rule entry, in seconds, in a request or assigned lifetime of this entry, in seconds, in a response . The value 0 indicates "delete".

Mapping Nonce: Random value chosen by the PCP client. Mapping Nonce MUST be copied and returned by the PCP server in a response.

TS Format: An 8-bit unsigned integer indicating the Traffic Selector Format. Value "0" is reserved and MUST NOT be used. The values for that field are defined in Section 3 of [RFC6088] and are repeated here for completeness.

* When the value of the TS Format field is set to (1), the format that follows is the IPv4 binary traffic selector specified in Section 3.1 of [RFC6088].

* When the value of the TS Format field is set to (2), the format that follows is the IPv6 binary traffic selector specified in Section 3.2 of [RFC6088].

Direction:

* 0 indicates outbound direction for traffic selector rule.

* 1 indicates inbound direction for traffic selector rule.

* 2 indicates inbound and outbound direction for traffic selector rule.

Reserved: 16 reserved bits, MUST be sent as 0 and MUST be ignored when received.

Traffic Selector: The traffic selector defined in [RFC6088] is mandatory to implement.

3.2. Generating a TSELECT Request

The PCP client, first does all processing described in Section 8.1 of [RFC6887]. It then includes the TSELECT OPCODE.

The Mapping Nonce value is randomly chosen by the PCP client, following accepted practices for generating unguessable random numbers [RFC4086], and is used as part of the validation of PCP responses by the PCP client, and validation for mapping refreshes by the PCP server.

The PCP client MUST use a different mapping nonce for each PCP server it communicates with, and it is RECOMMENDED to choose a new random mapping nonce whenever the PCP client is initialized. The client MAY use a different mapping nonce for every mapping.

3.3. Processing a TSELECT Request

The PCP server performs processing in the order of the paragraphs below.

Upon receiving a PCP request with the TSELECT opcode, the PCP server performs the processing described in Section 8.2 of [RFC6887]. If the PCP server can accommodate the request as described in the TSELECT request, it sends a PCP response with the SUCCESS response else returns a failure response with the appropriate error code.

Discussion: How to deal with overlap in traffic selector rules ?

3.4. Processing a TSELECT Response

Upon receiving a TSELECT response, the PCP client performs the normal processing described in Section 8.3 of [RFC6887].

4. IANA Considerations

In order to identify TSELECT Opcode, a new value (TBD) needs to be defined in the IANA registry for PCP Opcodes.

5. Security Considerations

Only certain users or certain applications can be authorized to signal TSELECT request. This authorization can be achieved using PCP authentication [I-D.ietf-pcp-authentication]. PCP authentication must be used for mutual authentication between the application signaling TSELECT request and the PCP-aware firewall. Without this authentication enabled, the TSELECT request is open for attacks with fake applications falsely claiming to be legitimate applications that require special treatment, i.e., the firewall infrastructure is at risk of being misused.

Should the firewall be spoofed, applications could be misled that the firewall has successfully processed the PCP request.

6. Acknowledgements

Thanks to Dan wing for valuable inputs and comments.

7. References

7.1. Normative References

[I-D.ietf-pcp-authentication]
Wasserman, M., Hartman, S., Zhang, D., and T. Reddy, "Port Control Protocol (PCP) Authentication Mechanism", draft-ietf-pcp-authentication-06 (work in progress), October 2014.

[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.

[RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, April 2010.

[RFC6088] Tsirtsis, G., Giarreta, G., Soliman, H., and N. Montavont, "Traffic Selectors for Flow Bindings", RFC 6088, January 2011.

[RFC6887] Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, April 2013.

7.2. Informative References

[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.

[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.ripke-pcp-tunnel-id-option]
Ripke, A., Dietz, T., Quittek, J., and R. Silva, "PCP Third Party ID Option", draft-ripke-pcp-tunnel-id-option-02 (work in progress), October 2014.

[RFC4086] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

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

Mohamed Boucadair
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange.com