

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 28, 2008

H. Soliman
Elevate Technologies
G. Daley

S. Krishnan
Ericsson
February 25, 2008

Firewall Control for Public Access Networks (FCON)
draft-soliman-firewall-control-01

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on August 28, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2008).

Internet-Draft

Firewall Control Protocol

February 2008

Abstract

This document proposes a new mechanism that allows end nodes to signal its preferences for traffic filters to a firewall function in the network or to another node that controls the firewall function in the network.

Table of Contents

1.	Requirements notation	4
2.	Introduction	5
3.	Protocol operation	7
4.	Firewall Scenarios	9
5.	PDP Discovery	12
5.1.	DHCP extensions	12
5.1.1.	PDP Sub-Option Format	13
6.	Authorization Mechanism	15
6.1.	Proof of ownership	15
6.2.	Firewall request policy	16
7.	Protocol Messages	17
7.1.	The Request Message Format	17
7.2.	The Response Message Format	18
7.3.	The Init Message	19
7.4.	The Init Acknowledgement Message	20
7.5.	Protocol Options	21
7.5.1.	The Acknowledgement Option	21
7.5.2.	The Filter Identifier Option	22
7.5.3.	The Nonce Option	23
7.5.4.	The Timestamp Option	24
7.5.5.	The IP Address Option	25
7.5.6.	The Cookie Option	27
7.5.7.	The Public Key Option	28
7.5.8.	The Lifetime Option	29
7.5.9.	The Certificate Option	30
7.5.10.	The Digital Signature Option	31
8.	Establishing a Secure Connection	34
9.	Creating New entries	35
10.	Updating Entries	37
11.	Requesting an IPv4 Address	38
12.	Timeouts and Retransmissions	39
12.1.	Session Start Delays	39
13.	IANA Considerations	40

14.	Security Considerations	41
15.	Acknowledgements	42
16.	Normative References	43
Appendix A.	Dynamic PDP Discovery (Informative)	45
	Authors' Addresses	47

Soliman, et al.

Expires August 28, 2008

[Page 2]

Internet-Draft

Firewall Control Protocol

February 2008

	Intellectual Property and Copyright Statements	48
--	--	--------------------

1. Requirements notation

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](#)].

[2.](#) Introduction

The need for network protection has led operators to deploy firewalls with typical configurations that limit the traffic coming into or out of a network to the administrators configuration. An administrators configuration is typically static and affects all users within the network. While such security measure is regarded as successful and is widely deployed, it has several drawbacks from a user and network operator's points of view. The assumption in such deployments is that there is a common configuration that satisfies all users. That is, the same type of applications are used by all network users and that all users do not need to be reachable unless they initiate a connection. Alternatively, in some deployment scenarios, e.g. IP Multimedia System (IMS) in 3GPP, the assumption is that externally initiated connection will use a signaling protocol, through local proxies, and therefore a local firewall can be manipulated by such proxies to allow such connections. Another assumption in deploying firewalls is that there is a Trusted side (the administrator's network) and an Untrusted side (The Internet). This is reasonable in a network where you assume no one on the inside of the network would intentionally try to attack the network infrastructure.

The assumptions above are clearly workable in a network where there is a defined behaviour for the users. For instance enterprise network users can easily be told what applications and protocols are acceptable for the users according to company policy. However, in a public network, such restrictions cannot be made without introducing serious limitations on users. Furthermore, in a public network, one can no longer assume that users on the inside are trusted. This is especially true in networks with anonymous users as well as users who may not be competent in protecting their own infrastructure.

The need for firewalling traffic still exists in public networks. Operators may wish to provide general protection for their infrastructure and users. At the same time, there is a clear need for flexibility in order to allow users a high degree of freedom in choosing the applications and protocols that they want to run. The firewall function may also be located closer to the user than the traditional deployment of firewalls on the core edge of the network. This allows for greater scalability by allowing the firewall function to serve a smaller number of users, which reduces bottlenecks in the core. This is particularly useful for IPv6 networks where there should be no need for Network Address Translators.

To allow for a more flexible firewall deployment, both in the location and configuration of the firewall, this document proposes a new mechanism that allows end nodes to signal its preferences for traffic filters to a firewall function in the network or to another

node that controls the firewall function in the network. The mechanism introduced in this document allows for a high degree of flexibility and security. A generic mechanism for firewall control has the following advantage:

- o Removes the reliance on specific application proxies for signalling to allow incoming connections.
- o Allows easy deployment of new services without requiring specific application gateways.
- o Allows easy deployment of new protocols that may not be known to the firewall and would otherwise be blocked.
- o Allows maximum user control, which provides the finest granularity

for firewall configuration.

One of the approaches for firewall control can be found in [\[I-D.ietf-nsis-nslp-natfw\]](#). However, this approach assumes a form of pre-established trust. With mobility becoming the norm on the Internet, it is difficult to assume trust without the use of global Public Key Infrastructure (PKI), which is not available today. Other approaches to firewall control include the presence of application proxies that are either colocated or physically separate from the firewall. The mechanism introduced in this document allows end nodes to signal their preferences directly to the firewall, or to another entity that controls the firewall. Hence, this mechanism allows network operators to continue the use of application proxies that control the firewall, while adding this new mechanism as a generic option.

[3.](#) Protocol operation

The FCON protocol allows nodes to send its preferences to a Policy Decision Point (PDP) in order to configure a firewall with those preferences. The PDP may be located anywhere in the network, including the access router sharing the node's link. Furthermore, it may be colocated with the firewall. FCON allows the node to create new entries, add entries, as well as, modify or delete existing

entries. FCON uses ICMP for transporting its messages between the node in question and the PDP. FCON can also be used to request the allocation of a unique IPv4 address and one or more port numbers by a Network Address Translator (NAT).

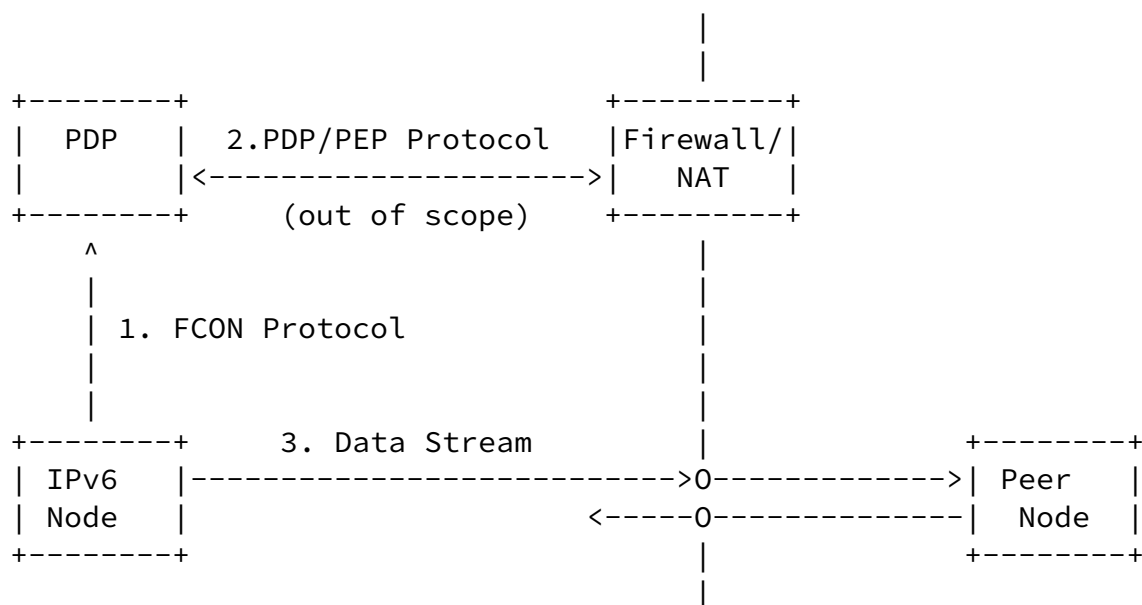


Figure 1: Communication architecture

Nodes that implement FCON first need to discover the IP address of the PDP in order to send future messages. The PDP's address is either included in a router advertisement option or a new DHCPv6 option. Alternative discovery mechanisms are described in [Appendix A](#). Following the discovery phase, a node may start sending messages to the PDP requesting that certain protocols or port numbers be opened for its traffic on a firewall or Policy Enforcement Point (PEP). A node may also request a unique IPv4 address and one or more port numbers for NAT traversal. All entries created by a node have a lifetime associated with them and MUST be refreshed in order to avoid losing them. The lifetime is set by the PDP and cached by the node using FCON.

Depending on the network configuration, from the end host's point of view, the PDP may be colocated with a firewall or separated.

Moreover, both functions could be colocated with an access router or

located within the core network. The protocol between the PDP and the firewall is outside the scope of this document. Authentication and authorization of FCON messages takes place by the PDP. The PDP is assumed to be authenticated with the firewalls through one of several methods, including manual configuration of security associations, public key-based authentication, or any other method deemed appropriate by the network administrator.

Message authentication and authorization is a critical component of the FCON protocol. Different deployment models may have different requirements for authentication and authorization. In some deployments the use of public keys and trusted certificates can be sufficient to authorize an end node for FCON messages. Examples of such deployments may include enterprise or home networks. Other deployments may require proof that the sender is authorized to perform the action requested. Given the information exchanged in the FCON messages, it is sufficient for a node to prove the ownership of the address included in a message to be authorized to perform the requested action provided that such action does not violate the network administrator's policies. Hence, the PDP first needs to know that a sending node owns the address included in the message, then ensure that the request does not violate any known policies. Following such verification the PDP would configure the firewall/NAT with the necessary information requested by the end node.

This specification allows FCON security to be based on self-generated public keys and Cryptographically Generated Addresses (CGAs). This allows nodes to provide the necessary address ownership credentials in those deployments that require it. Alternatively, public keys associated with PKI can be used for deployments that do not require proof of address ownership.

4. Firewall Scenarios

This section describes limitations in current firewall deployment, and illustrates challenges in attempting to control firewalls dynamically using FCON. Scenarios are described which motivate features described in this document.

Manual policy firewalls describe the state of the art, with respect to deployed networks. A management platform acts as a static policy decision point, propagating policy out to one or more policy enforcement points. Such an environment is displayed in Figure 2. Connection state creation occurs when data plane packets compliant with policy are received.

Static firewalls such as these may exist where the PDP and PEP are collapsed into the same entity, and long term configuration occurs on the same device where connection state is created.

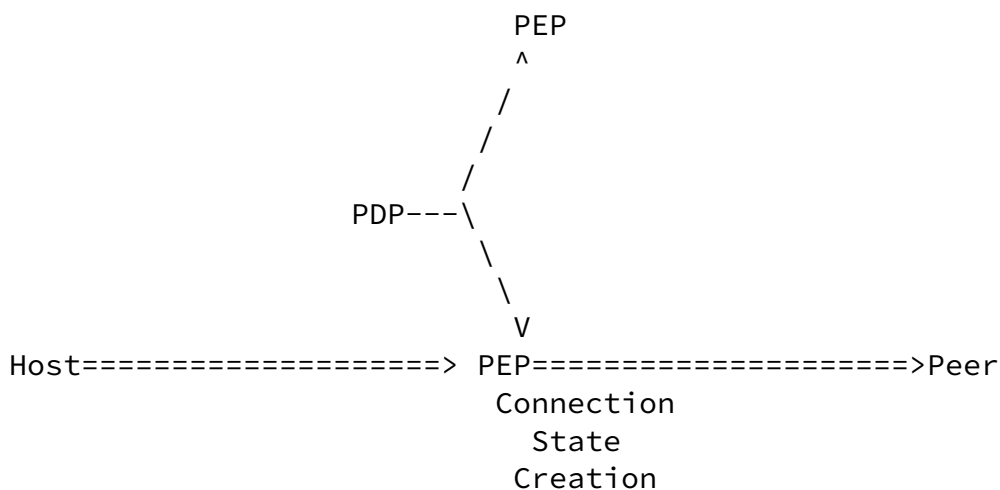


Figure 2: Static Firewall

As shown in Figure 2, where connection state is required for inbound packet flows, the PEP's preconfigured policy is used to create inbound connection state. Such connections are available at all times, and depend upon the specificity of the PDP's inbound policy in order to maintain internal network security. These inbound communications remain available regardless of the lifetime of the individual server applications.

Internet-Draft

Firewall Control Protocol

February 2008

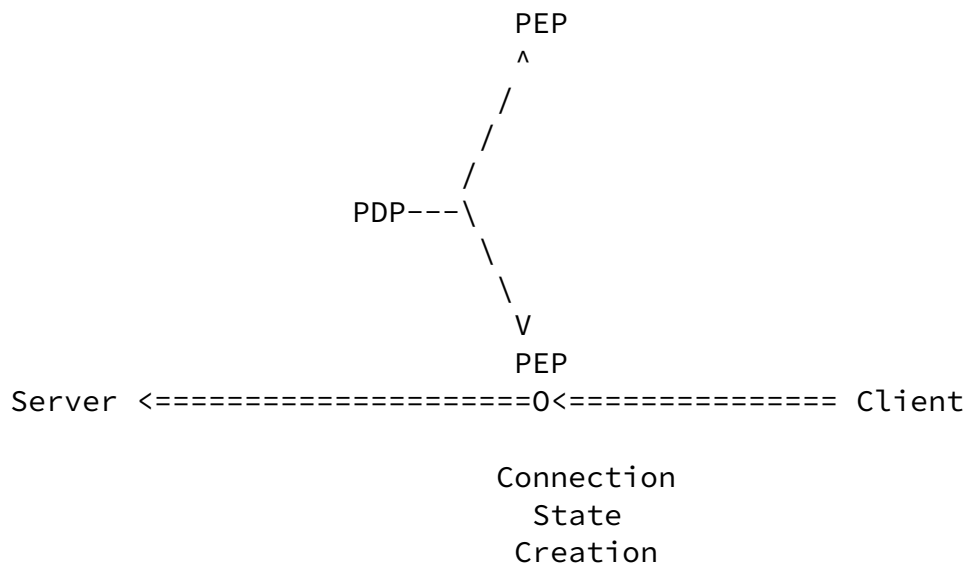


Figure 3: Inbound Connection on Static Firewall

Problems emerged when people attempted to support peer-to-peer applications with dynamic source and destination addresses. This is illustrated in Figure 4. Transport layer connection state is unable to identify the upper layer inbound connection associated with the peer-to-peer application. Inbound packets are dropped on the secondary session unless the Firewall snoops and understands the specific protocol, and the inbound policy is loosened to permit such inbound sessions.

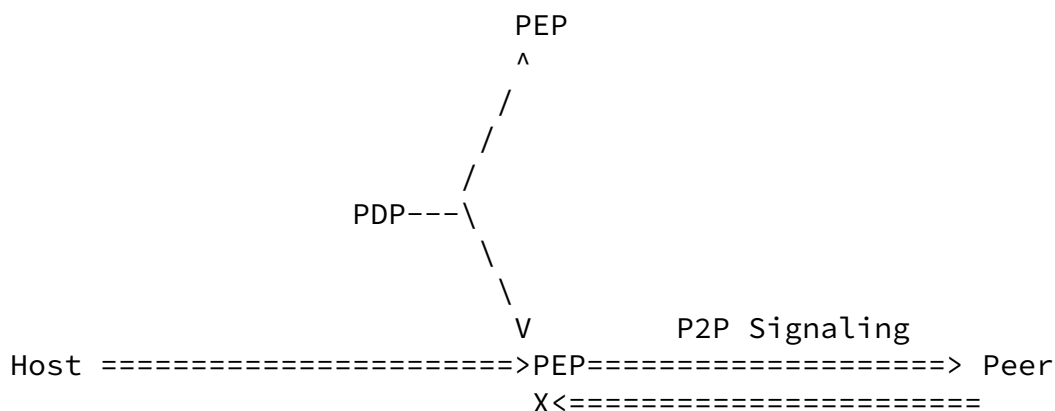


Figure 4: Peer-to-Peer Connection on Static Firewall

By enabling dynamic signalling, a host can inform the network elements of its intention to send packets with a particular source and destination address, and transport profile. This means that the policy decision and enforcement can occur at the time that the session is established, and adapt specifically to the needs of the network's current transmission profile.

Soliman, et al.

Expires August 28, 2008

[Page 10]

Internet-Draft

Firewall Control Protocol

February 2008

In Figure 5, the host informs the PDP which protocols are expected through the network gateways using a Request message to create a flow descriptor entry in the firewall. The PDP indicates whether the session will be allowed, and state created on the PEP. In this case, a Response message is sent, indicating that the PDP believes the communications are allowed.

```
Request
/-----PDP..
//-----/
// ACK .
/ V V
Host =====>PEP===== > Dest
```

Figure 5: Outbound Session with PDP Signaling

In many environments, multiple Firewalls exist on the path to the Internet. This is shown in Figure 6.

Signalling is either performed to a single PDP which passes information to both PEPs, or to a separate PDP for each PEP. By using separate PDPs, different trust policy and vendor independence may be readily achieved. This may particularly be applicable where the internal firewall is operated by an enterprise, and the exterior firewall by a service provider.

```
Request
/----->PDP..
//-----/
```

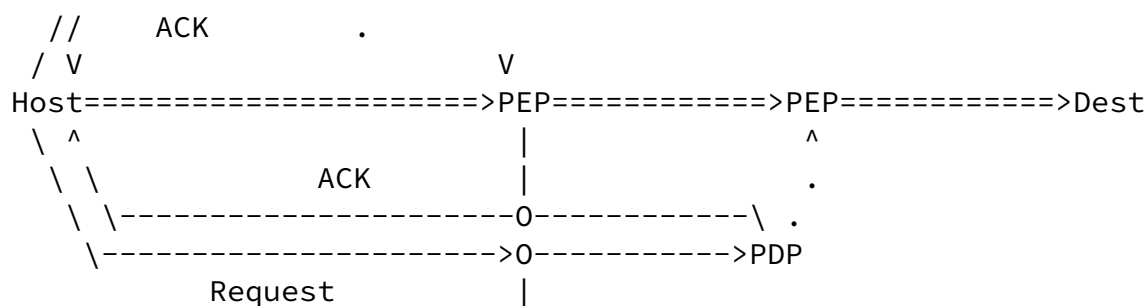


Figure 6: Multiple On-Path Firewalls

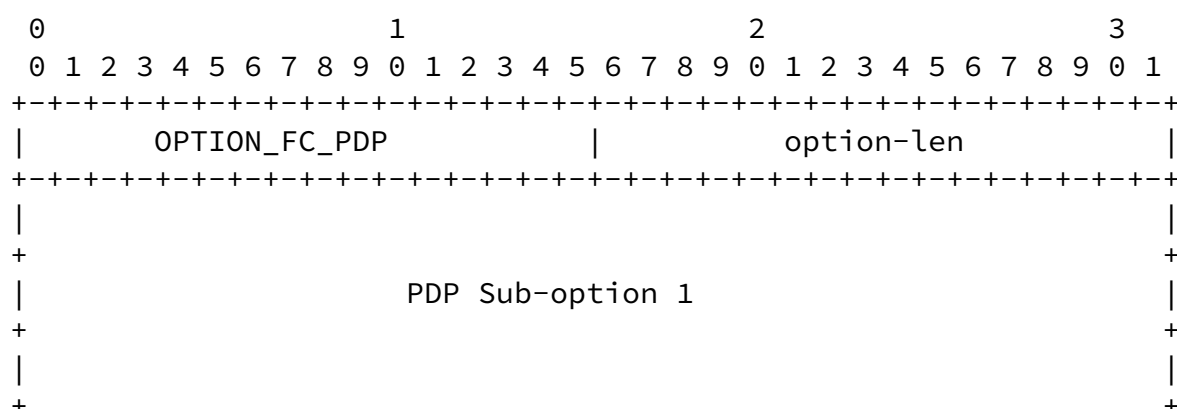
5. PDP Discovery

5.1. DHCP extensions

In order to configure PDPs on hosts using the existing access infrastructure, a DHCP option is provided.

The option format includes PDPs with destination coverage information on a per-PDP basis in a suboption format. This would allow for specific PDPs to have jurisdiction over different network ranges (for example, for a Data Centre, and an Internet Firewall).

The DHCP option for identifying PDPs is described using the format:



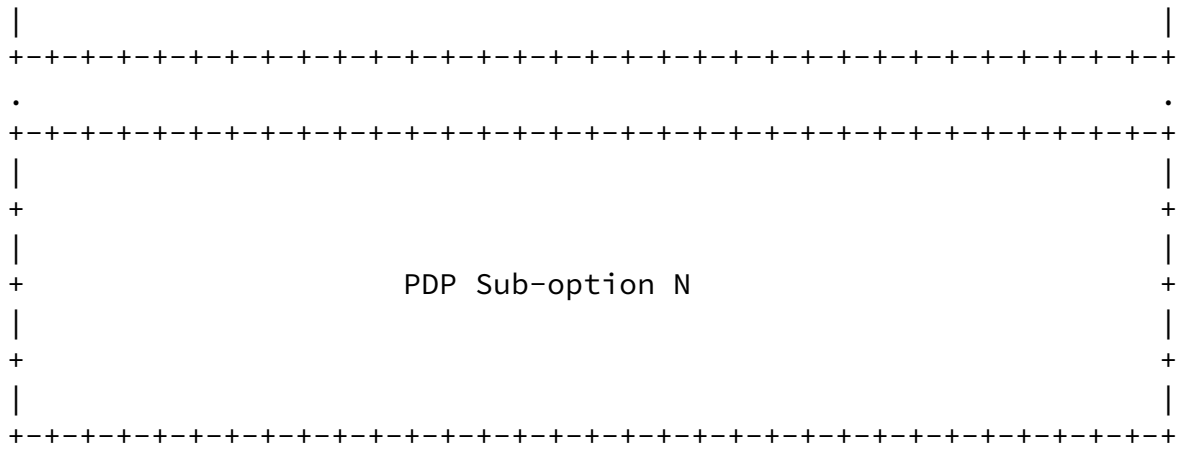


Figure 7: FCON PDP Discovery DHCPv6 Option

OPTION_FC_PDP

Assigned DHCP option code for PDP discovery. TBD.

option-len

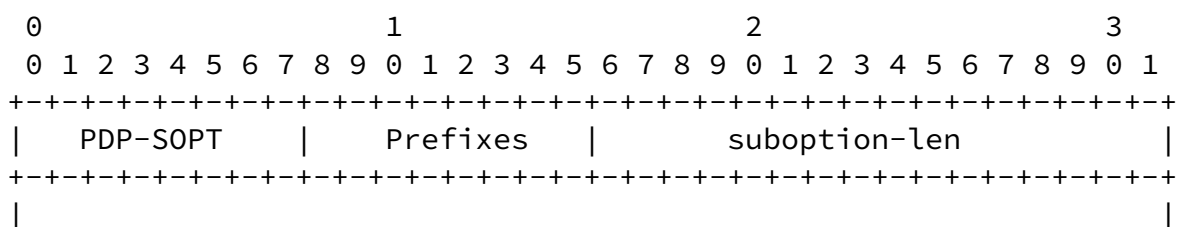
The length of the entire option in bytes, less 4

PDP Sub-option

A sub option containing information for each PDP the host should configure.

[5.1.1.](#) PDP Sub-Option Format

Each PDP has its own IP address and validity information.



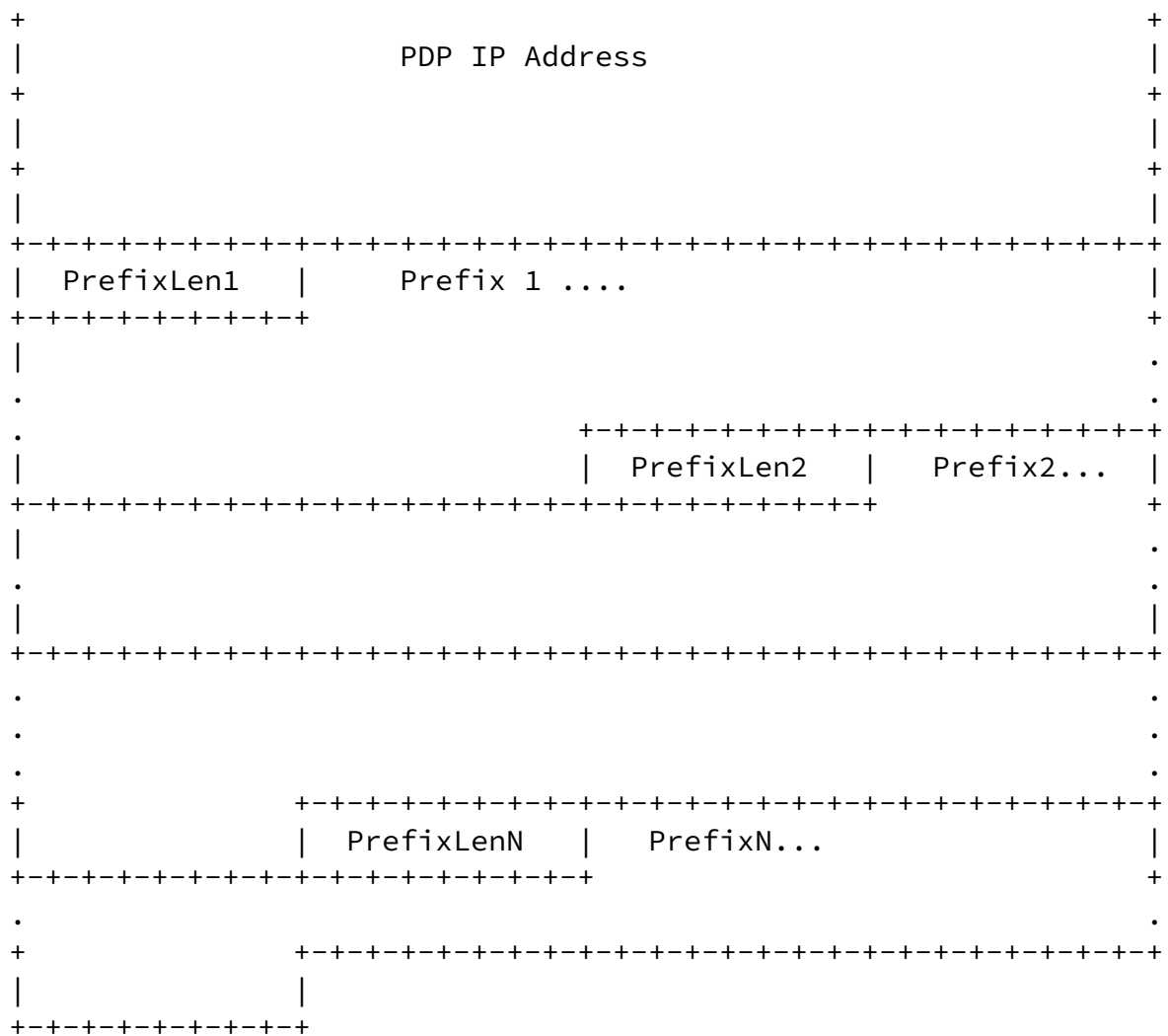


Figure 8: FCON PDP Discovery DHCPv6 Option

PDP-SOPT

A sub option

Prefixes

The number of prefixes covered by this PDP.

suboption-len

The length of the PDP sub-option in bytes, including all fields.

PDP IP Address

The IP Address of the PDP which is used as a destination for FCON requests.

PrefixLen i

The length of the next prefix, in bits.

Prefix i

The space consumed by the Prefix is the number of bytes required to store the prefix (PrefixLen i/8), rounded up to the nearest integer. A Prefix of 0::/0 is encoded with a zero length Prefix Field, and indicates the PDP is responsible for all destinations.

[6.](#) Authorization Mechanism

The protocol described in this document uses a two step authorization procedure. The right to control the firewall will be granted if

- o The node making the request is behind the firewall and owns the IP address
- o The request made by the node is allowed by local firewall policy.

6.1. Proof of ownership

This document relies on Cryptographically Generated Addresses (CGAs) as defined in [[RFC3972](#)] in order to prove that the requesting node owns the address from which the firewall control request was sent. This is possible because the CGA address of the node is derived based on a hash of the public key of node with other known pieces of information like the prefix. The procedure for verification of CGA addresses is described in [Section 5 of \[RFC3972\]](#). If the PDP receive a signed firewall control request message that includes the public key and the CGA of the requester it can verify that the sender of the request indeed owns the address and that the address corresponds to the public key carried in the message. Since creating this signature requires the corresponding private key of the public key contained in the message, it can also conclude that the message has not been tampered with. In addition to this, to prevent replay attacks, the PDP can verify that the sender is in fact reachable and alive, using a to be defined FCON protocol message that uses nonces to verify that the received message was not replayed from an earlier run of the protocol. e.g. The PDP could send a nonce to the requesting node in this message encrypted by the public key of the requesting node. Only the requesting node can decrypt this message and obtain the nonce. Then it needs to increment the nonce and encrypt it and send it back to the PDP. Once the PDP receives this message it can be assured that the requester is alive and reachable. If either of these tests fail, the PDP rejects the firewall configuration request with an error that indicates that the address ownership was not confirmed.

Please note that the PDP does not have to verify who owns the public key. It just needs to verify whether the owner of the address is the same as the owner of the public key contained in the signed message. It can do so solely based on the information contained in the message prefix of the address, public key etc., by running the algorithm mentioned earlier.

[6.2.](#) Firewall request policy

Once the PDP has verified that the requesting node owns the concerned address and is reachable, the PDP needs to start processing the request message itself. Since it is possible that the firewall control request may not conform to administrative policy, the PDP verifies that the request falls within the parameters specified by such policy. If it does, the PDP starts acting on the request and sends configuration messages to the necessary firewall(s). If not, the PDP rejects the firewall configuration request with an error that indicates that the request did not comply with local administrative policy.

7. Protocol Messages

FCON uses ICMPv6 for transporting its messages. The protocol is based on a request response message exchange. Hence, two ICMP message types are needed. The ICMP Code field is used to distinguish different types of FCON messages. Each message can contain one or more options. This sections lists all protocol messages and options.

7.1. The Request Message Format

The Request message is sent from the end node to the PDP. The purpose of the message depends on the options included. For each operation described in this specification a specific set of options must be included. The format of the request message is shown below.

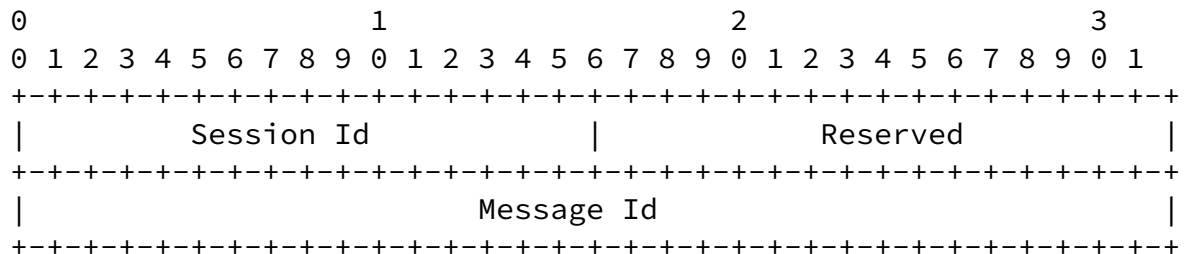


Figure 9: The Request Message Format

Session Id

An unsigned 16-bit integer that includes a session identifier. The session identifier is initially picked by the end node when a security association is created with the PDP. The Session Id **MUST** be unique for a particular end node, which is identified by its public key.

Reserved

This field **MUST** be set to zero by the sender and ignored by the receiver.

Message Id

This field include a message identifier. The message identifier is a simple counter incremented by 1 for every new message. This field is used to match responses with their correponding requests.

7.2. The Response Message Format

The Response message is sent from the PDP to the end node in response to a request message sent from the end node. The response message includes the same Session Id and Message Id that were included in the sender's Request message. The Response message may contain several options. The options contained in a given message depend on the type of operation requested in the Request message. The format for the Response message is shown below.

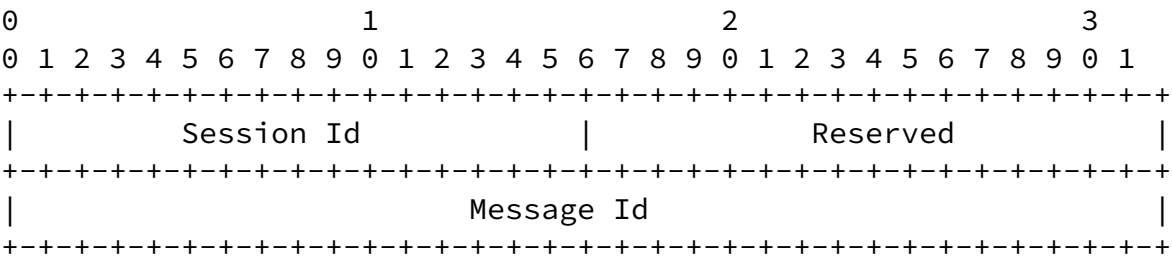


Figure 10: The Response Message Format

Session Id

An unsigned 16-bit integer that includes a session identifier. The session identifier is initially picked by the end node when a security association is created with the PDP. The Session Id MUST be unique for a particular end node, which is identified by its public key.

Reserved

This field MUST be set to zero by the sender and ignored by the receiver.

Message Id

This field include a message identifier. The message identifier is a simple counter incremented by 1 for every new message. A Response message's identifier is set to the message identifier of the corresponding request message.

7.3. The Init Message

The Init message is sent from the end node to the PDP in order to establish a secure association before sending further requests. This message MUST NOT include information about flows that need to be installed in the firewall. Instead, the message contains the end node's security credentials and a challenge for the PDP.

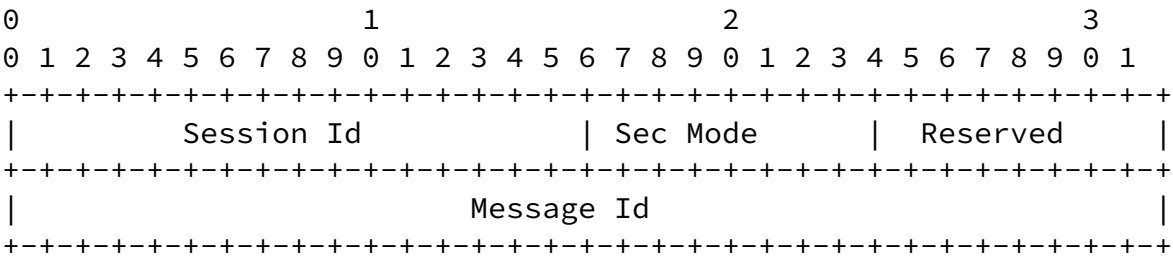


Figure 11: The Init Message Format

Session Id

An unsigned 16-bit integer that includes a session identifier. The session identifier is initially picked by the end node when a security association is created with the PDP. The Session Id MUST be unique for a particular end node, which is identified by its public key.

Sec Mode

This field indicates the type of credentials used to establish the security association. A value of 1 indicates the use of self-generated public keys. A value of 2 indicates the use of trusted certificates, which are either signed by the same administrative authority or a trusted third party.

Reserved

This field **MUST** be set to zero by the sender and ignored by the receiver.

Message Id

This field include a message identifier. The message identifier is a simple counter incremented by 1 for every new message.

[7.4.](#) The Init Acknowledgement Message

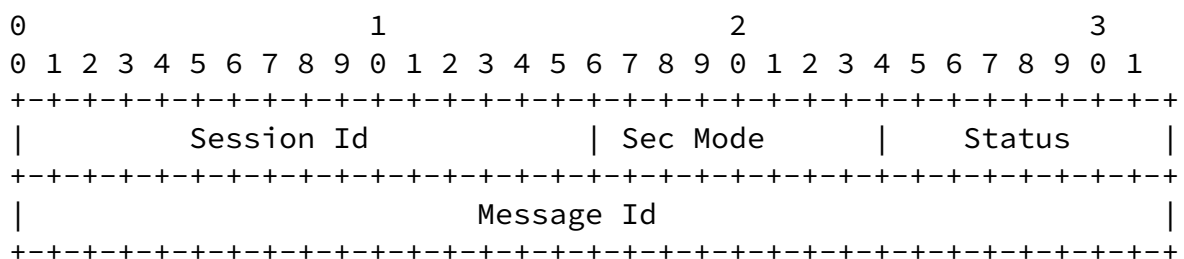


Figure 12: The Init Acknowledgement Message Format

Session Id

An unsigned 16-bit integer that includes the session identifier included in the Init message.

Sec Mode

This field includes the same value for the Sec Mode field included in the Init message if the Status field indicated a successful operation. Otherwise, the field includes the value supported by the PDP.

Status

This field indicates the success or failure of the processing of the Init message. Values below 128 indicate success, while values of 128 and above indicate failure.

Message Id

This field includes the value of the Message Id that was included in the Init message being responded to.

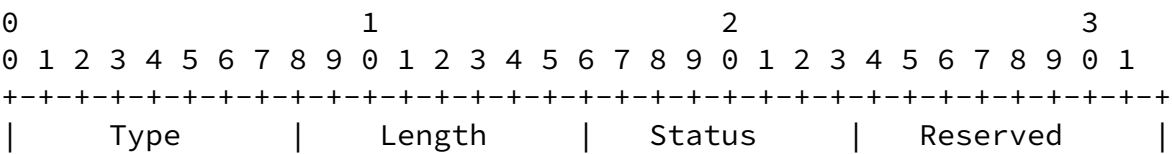
The following values are reserved for the Status field:

- 0 Success
- 128 Reason unspecified
- 129 Security mode not supported
- 130 Invalid format
- 131 Certificate not accepted

7.5. Protocol Options

7.5.1. The Acknowledgement Option

The Acknowledgement Option is used to carry information about a requested operation. The format of the Acknowledgement option is shown below.



The filter identifier option is used to encode information that describes a flow and the treatment needed for such flow. A host may request that a flow be allowed, blocked, or removed from the firewall, which defaults to the firewalls original settings.

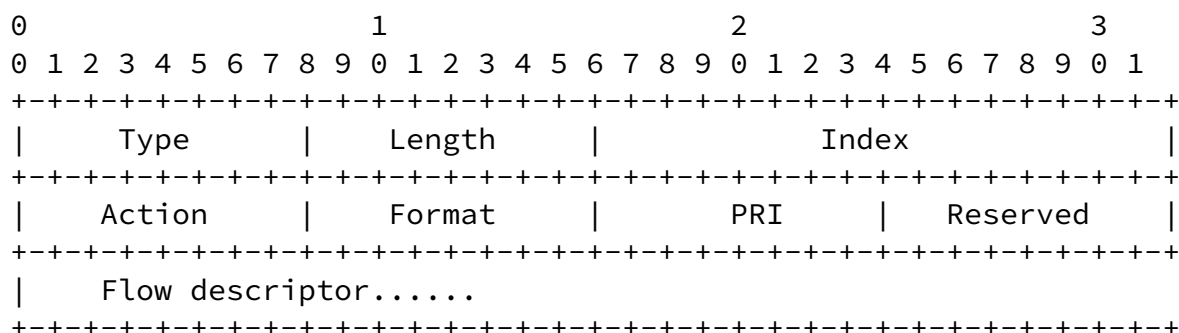


Figure 14: The Flow Identifier Option Format

Type

A value assigned to this option. TBA

Length

The length of this option in 8-octet units.

Index

A Unique value that identifies a particular flow description. This value is assigned to the flow description by the end node.

Action

This field indicates the type of operation requested by the end node for the flow included in the option. The following values are assigned to this field. A Value of 1 indicates a request to Allow the flow. 2 indicates a request to Block a flow. 3 indicates a request to Update a flow. 4 indicates a request to Delete a flow.

Format

This field indicates the format used for the flow descriptor.
Values TBA

PRI

This field indicates the priority of a particular flow. A lower value indicates a higher priority. A value of 1 indicates the highest priority. A value of zero is not allowed by this specification. The priority field is needed in cases where two flow descriptions overlap while having conflicting Action fields. The Action field included in the option with higher priority takes precedence.

Reserved

This field MUST be set to zero by the sender and ignored by the receiver.

7.5.3. The Nonce Option

The Nonce Option is illustrated in Figure 15 and is used to ensure freshness in acknowledgements from the PDP to the requesting FCON node. It works by making sure that the PDP copies the Nonce Option from the request into the response. The Nonce is checked along with other freshness and authentication information before

As such, a new Nonce MUST be selected for each transmission of a request message. PDPs receiving a valid request message MUST copy the Nonce option into the response, regardless of the status of any individual flow.

The Nonce MUST be unpredictable, and SHOULD contain hardware generated randomness, where possible.

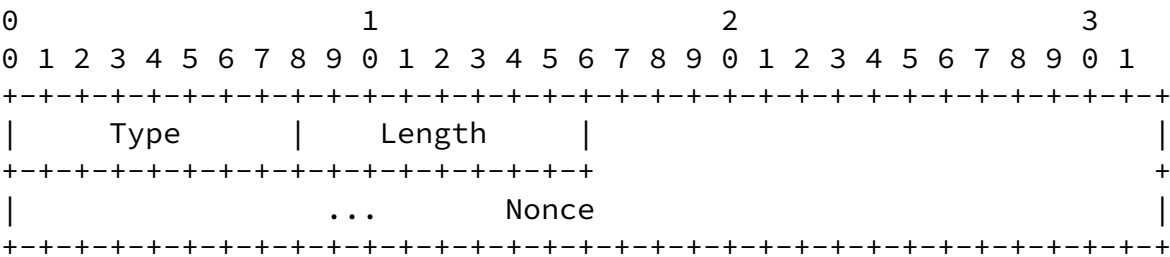


Figure 15: The Nonce Option Format

Type

Internet-Draft

Firewall Control Protocol

February 2008

A value assigned to this option.

Length

The length of this option in 8-octet units.

Nonce

The Nonce is a block of data selected by the requester to be sent to the PDP. the entire length of the Nonce block MUST NOT exceed 384 bits. It MUST contain at least 64 bits of unpredictable random data. It SHOULD contain significantly more randomness.

PDPs receiving the Nonce SHOULD cache it to ensure that duplicate request packets are not processed from the same source.

[7.5.4.](#) The Timestamp Option

The Timestamp option format is used to ensure attackers are unable create state in the future by replaying signed FCON messages. It relies on congruence between clock information within nodes and PDPs, and therefore has some limitations where secured time synchronization is not available

The following option format follows the principles and message formats as described in the [[RFC3971](#)], except that its protections may be weaker when operating in a multi-hop domain.

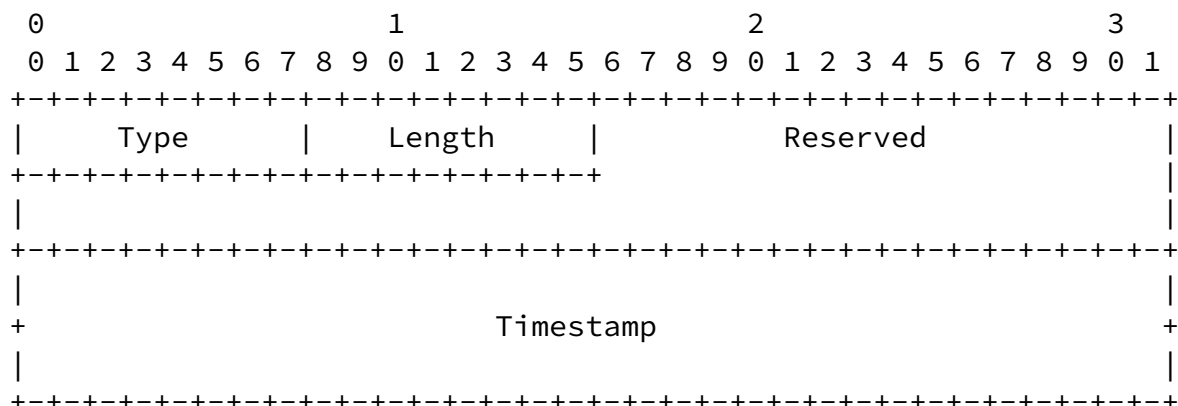


Figure 16: The Timestamp Option Format

Type

A value assigned to this option.

Length

Soliman, et al.

Expires August 28, 2008

[Page 24]

Internet-Draft

Firewall Control Protocol

February 2008

The length of this option in 8-octet units.

Timestamp

A 64-bit unsigned integer field containing a timestamp. The value indicates the number of seconds since January 1, 1970, 00:00 UTC, by using a fixed point format. In this format, the integer number of seconds is contained in the first 48 bits of the field, and the remaining 16 bits indicate the number of 1/64K fractions of a second.

Each FCON message MUST contain the Timestamp option, and recipients SHOULD ensure that Timestamps are valid to prove freshness of the message.

Processing of this option follows that for SEND, except that the corresponding figures for clock drift and fuzz are more lenient. This allows for longer attack windows of attack against FCON infrastructures, but also allows for deviations in packet transfer delays on multi-hop networks and the extended duration of the state created in Firewalls, as opposed to Neighbour Discovery:

TIMESTAMP_DELTA 600 seconds (10 minutes)

TIMESTAMP_FUZZ 6 seconds

TIMESTAMP_DRIFT 1 % (0.01)

[7.5.5.](#) The IP Address Option

The IP address option is used by a node to request a unique IPv4 address and one or more port numbers.

Internet-Draft

Firewall Control Protocol

February 2008

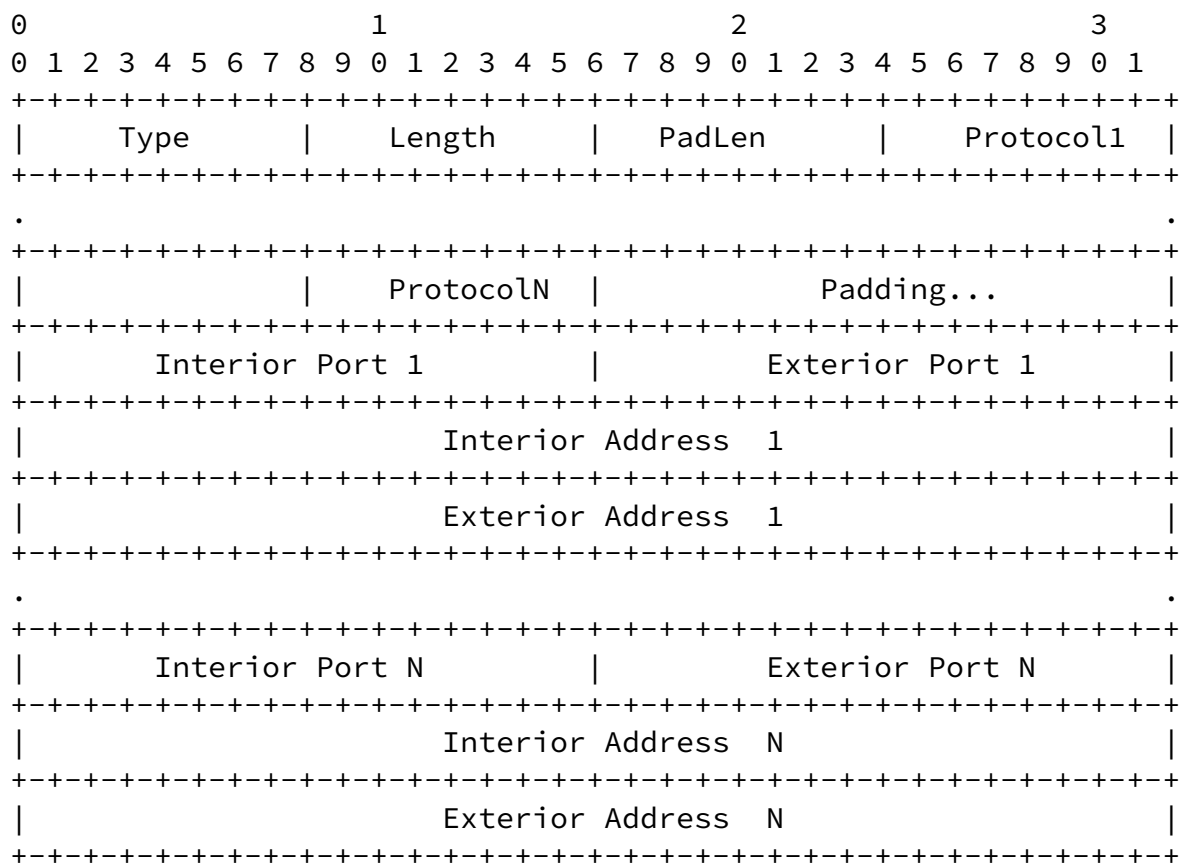


Figure 17: The IP Address Option Format

Type

A value assigned to this option.

Length

The length of this option in 8-octet units

Padlen

The length in bytes of the Padding field

Protocol i

The ith IP protocol number

Padding

This field SHOULD be set to zero by the sender and ignored by the receiver.

Interior Port i

The 16 bit Transport layer internal identifier of the of the ith session for which a mapping is requested. Where such identifiers do not make sense for a particular protocol, this field SHOULD be set to zero

Exterior Port i

The 16 bit Transport layer identifier of the of the ith session for which a mapping is requested. Where such identifiers do not make sense for a particular protocol , this field SHOULD be set to zero Requesters SHOULD set this field to zero, if they wish the PDP to assign a port

Exterior Address i

The address of the ith external IP address to which a mapping is requested. Requesters SHOULD set this field to zero, if they wish the PDP to assign an address.

Interior Address i

The address of the *i*th internal IP address for which a mapping is requested

7.5.6. The Cookie Option

The Cookie Option contains a string of information chosen by the PDP to the host to when requesting further security credentials.

The cookie can contain any information which the PDP desires, and the cookie must be returned to the PDP along with credential information

When the host sends further credential information it MUST add the cookie to the Init or SEND Certification Path Advertisements sent to validate the credentials.



Figure 18: The Cookie Option Format

Type

The assigned type of this option (TBD).

Length

The length of this option in 8-octet units.

C

The Flag indicating Certificate Path Discovery is ok (Flag Set).

This flag being set allows the response to use [\[RFC3971\]](#) Certification Path Advertisement Messages to convey the list of

certificates for the respondent. If this flag is not set, the sender **MUST** use Init messages instead.

This Cookie is required to appear in all such messages (IANA Note).

Length

A string of bytes chosen by the PDP to ensure liveness of responses. The entire option, carrying this field needs to be copied into an INIT for transmission to the PDP, along with credential information.

7.5.7. The Public Key Option

The Public Key option is used to provide information to the PDP about the identity being used to sign the message. By using the key information in this option, or a cached copy, the PDP can use information in the Digital Signature Option, to verify the message's integrity.

This Option **MUST** be present in the message sent from a particular host to a PDP, and from a PDP to a host with which it has not communicated, unless the same information is provided within the message using a Certificate Option. If a host or PDP communicate with each other during a period where security state is still in existence, then the sender **MAY** leave this option out.

Where a receiving endpoint does not support a Key Type, it may indicate this to the far end in an acknowledgement but otherwise **SHOULD NOT** create any state in PEPs.

[illegible]

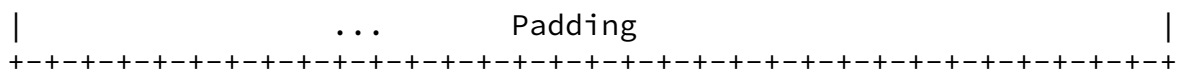


Figure 19: The Public Key Option Format

Type

A value assigned to this option. TBA

Length

The length of this option in 8-octet units.

Key Type

A description of the keying information to be supplied in the following Public Key Information field. The algorithms and formats to be supported are to be determined. A type value of 1 is CGA with the Public Key Information containing information compatible with [[RFC3972](#)] formats.

Pad Length

The length of the pad in bytes

Public Key Information

A stream of bytes describing a public key according to the algorithm specific format specified in the Key Type.

Padding

This field SHOULD be set to zero by the sender and ignored by the receiver.

[7.5.8.](#) The Lifetime Option

This lifetime option is included in the Response and Init Acknowledgement messages from the PDP to the end node to indicate the lifetime for the entries in a Request message or to indicate the

lifetime of a security association, respectively. The option format is shown below.

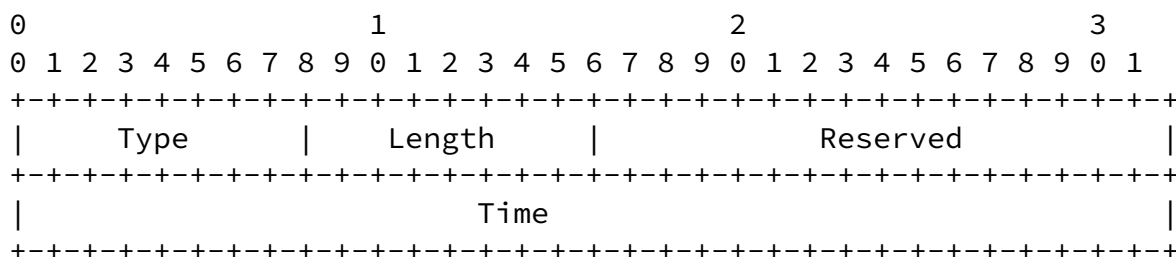


Figure 20: The Lifetime Option Format

Type

A value assigned to this option.

Length

The length of this option in 8-octet units.

Reserved

This field **MUST** be set to zero by the sender and ignored by the receiver.

Time

This field contains the lifetime in units of seconds.

7.5.9. The Certificate Option

The Certificate option contains a digital certificate issued by one of the Certificate Authorities in the sender's trust chain. It provides information about trust delegated to the sender or its authorizing authorities. This option follows the same format as in [\[RFC3971\]](#), and the certificates can be sent in Certification Path Advertisement messages, between hosts and PDPs.

The Certificate Option **MAY** be included in an FCON request or response message instead of including the Public Key option. It is **RECOMMENDED** that only one of the Public Key Option or Certificate Option is included in any FCON request or response message.

Internet-Draft

Firewall Control Protocol

February 2008

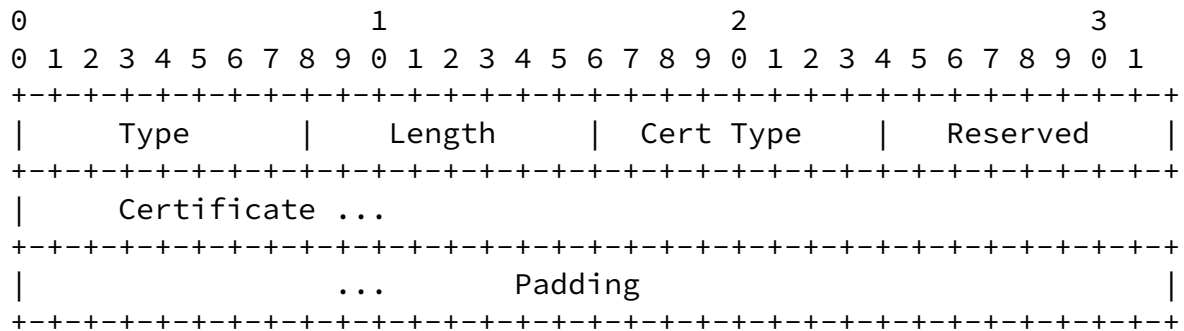


Figure 21: The Certificate Option Format

Type

A value assigned to this option.

Length

The length of this option in 8-octet units.

Cert Type

The type of certificate presented in the Certificate field. Type 1 is an X.509v3 digital certificate.

Certificate

A stream of bytes describing a one of the sender's certificates from it's trust chain. The format of a particular certificate is governed by the Cert Type.

Padding

This field SHOULD be set to zero by the sender and ignored by the receiver.

7.5.10. The Digital Signature Option

The Digital Signature Option MUST be included in every FCON message, and authenticates the preceding message contents. It does this by performing a signature over the message contents by the key

Type

A value assigned to this option.

Length

The length of this option in 8-octet units.

Sign Type

The type of digital signature used within the Digital Signature field of this option.

Type 1 is allocated as a PKCS 1.5 Digital Signature over the field sequence as described in [[RFC3971](#)].

Pad Length

The length of the Padding field in octets.

Key Hash

The left most (most significant) 128 bits of an SHA-1 hash over the public key information included in a Public Key Information field of a Public Key Option.

Digital Signature

A digital signature of the format specified in the field Sign Type. This signature is over the entire message, including the ICMPv6 Pseudo Header, and all options up to and preceding this option.

The length of the Digital Signature field is the length in octets of the message subtracting 12 octets for fixed length fields, and the contents of the Pad Length field.

Padding

This field SHOULD be set to zero by the sender and ignored by the receiver.

[8.](#) Establishing a Secure Connection

A host wishing to request that state be created on a PEP signals a PDP with either an Init message containing credential information, or a Request message containing the filter information describing the host's intended protocol behaviour.

Either of these messages MUST contain the Public Key Option (or a Substituted Certificate Option containing the relevant Public Key) and a Digital Signature Option, along with Timestamp and Nonce information. In addition, the Init message contains a unique session identifier field selected by the requesting node. This session identifier field will be used for subsequent signaling of other messages upon successful establishment of a secure session.

If the PDP can determine from the contents of the message that it believes the host is an acceptable requester, it can respond to the request with the appropriate success acknowledgement in the Init Acknowledgement message.

If the PDP is not able to authenticate or authorize the credentials for the host, it replies with an Init-Ack or Response containing a

non success response code, and a Cookie Option. Where the host wishes to establish a secure connection, the host must return this same cookie in the subsequent message (or messages for trust chains which exceed a single packet).

The host may then send additional information in the form of either a sequence of Certificate Path Advertisements, if allowed in the Cookie option. Finally, when all credentials are transmitted, the host again sends an Init message containing the last received cookie.

For the case where the Init-Ack contained the failure code "Liveness Test Needed", no further credentials are required, although an Init MUST be resent containing the received Cookie Option.

[9.](#) Creating New entries

Entries consist of one or more flow identification options that are sent from an end node to the PDP. In order to create entries the end node must send a Request message to the PDP. The Request message MUST contain an appropriate session identifier value, and a Message identifier. The message identifier can be implemented as a monotonously increasing counter. The Request message contains the following options:

- o One or more Flow Identification options. Each option contains a unique Index field, an appropriate Action field, a format field indicating the format of the flow descriptor and an appropriate priority in the PRI field.

- o The Nonce option
- o The Digital Signature option.

Upon receiving the Request message, the PDP verifies the signature in the message. If the verification fails, the message is silently discarded.

Upon successful verification of the Request message, the PDP checks the message header. If the session identifier is not known, the PDP sends a Response message with the same session and message identifiers and includes an Acknowledgement option with the status field set to 130.

If the message is correctly formed, the PDP inspects each flow identification option. If an error is found in any of the flow options, the PDP sends a Response message with the Acknowledgement indicating failure with the appropriate error code. The failed options MUST be included in the Response message. Successful options are processed by the PDP.

The process of updating existing flows is described in [Section 10](#).

If all options are processed successfully, the PDP sends a Response message. The session and message identifier fields are set to the same value in the Request message. The Response message includes the following options:

- o The Acknowledgement option. This option's status field indicates success
- o The Lifetime option. This option includes the lifetime associated with the new entries. The end node needs to refresh

those entries before the lifetime expires.

- o The nonce option.
- o The Digital Signature option.

Note that the process of adding new entries does not require the end nodes to send all existing entries. Only new flows need to be

included in the Request message.

Updating existing entries may take place due to the need for changing the flow description, deleting an existing entry, or simply refreshing an entry before its timer expires. When refreshing entries there is no need for the requesting node to send the entire Flow identifier option in a Request message. It is sufficient to send the option without the flow descriptor.

Updating entries is done using the same Request message as described in [Section 9](#). Upon receiving the Request message, the PDP verifies the Digital signature option. If the verification failed, the message is silently discarded.

After successfully verifying the message, the PDP processes each flow identifier option for formatting, flow descriptor (if the FID field indicates a new flow) and the Action field. If all of the flow identifier options are rejected, the PDP responds with a Response message, including the acknowledgement option, which indicates failure with an appropriate error code.

If some of the filter identifier options are rejected and others are accepted, the PDP responds with the Response message, including the acknowledgement option, which indicates partial success. The Response message also includes the lifetime option with an appropriate lifetime for the accepted options. In addition, the Response message includes all of the failed options. Each of the failed options includes a Status field, which indicates the reason for failure.

After receiving the Response message, the requesting node updates its list of accepted entries and the corresponding lifetimes for those entries.

11. Requesting an IPv4 Address

A requesting node may wish to know the public IPv4 address and port numbers allocated to it by a NAT for one or more of its applications. To do that, it can request one or more IP addresses and port numbers while specifying the protocol that it wants to use. This is done by sending a Request message that includes the following options:

- o The IP address option.
- o The nonce option.
- o The digital signature option.

Upon receiving the Request message, the PDP verifies the digital signature included. If the verification failed, the PDP silently discards the message.

If the PDP allows nodes to request an IPv4 address, it can proceed with the processing of the IP address option. Otherwise, the PDP rejects the request by sending a Response message with an acknowledgement option containing the appropriate error code.

If the IP address option is valid, the PDP proceeds to allocate the requested address(es) and port numbers. The method used by the PDP to communicate with the NAT/firewall is outside the scope of this document.

If after allocating the IP address to the requesting node, the PDP sends a Response message including the IP address option, which includes the allocated addresses and ports, the nonce option, the lifetime option and the digital signature option.

Upon receiving the Response message and verifying the digital signature option, the requesting node can use the IP address(es) and ports allocated in the message for the duration of the lifetime option. Should the requesting node wish to refresh the allocated addresses and ports, it MUST send the Request message with the IP address option including all the previously allocated IP addresses and ports.

[12.](#) Timeouts and Retransmissions

Where a host receives no response to a packet sent directly to a PDP, it may need to retransmit its initial packet. Each request MAY be transmitted up to four (4) times, each with a different Nonce and updated TimeStamp. Separation between one transmission and its next should be performed such that timeouts are exponential. RECOMMENDED timeouts are 1,2 and 4 seconds. A host SHOULD delay an initial retransmission by between 0 and 100ms, to ensure any retransmissions are serialized.

The PDP never sends packets to the host, except in response to an FCON message. In order to respond in sufficient time, it is recommended that the PDP respond without induced delay to any packet sent directly to its IP address.

[12.1.](#) Session Start Delays

When a new session is established, if a data plane transmission delays until FCON acknowledgements are received, there is likely to be significant added delay for every TCP or UDP session creation. It is therefore recommended that FCON immediately precedes packet transmission on the data plane, where it is not harmful to lose one or two of the initial packets. This may for example, be the case with unreliable protocols such as VoIP.

Also, when a host has one or more existing communications open in negotiation with a PDP, it is possible to send the packets immediately after sending the FCON request. This optimizes for the case where new state is able to be created immediately, and reduces latency at the risk of causing packet loss and retransmission on initial packets.

Where a host has not communicated to a PDP previously, it MAY induce additional delay before sending the data plane packets, in order to limit additional delays due to retransmission and timeout at the Transport Layer of the data session.

Significantly, when FCON packet loss or delay occurs on the

signalling path, this means that packet loss or delay will ensue, with the timings described in the prior section.

Soliman, et al.

Expires August 28, 2008

[Page 39]

Internet-Draft

Firewall Control Protocol

February 2008

[13.](#) IANA Considerations

TBD

[14.](#) Security Considerations

Certain assumptions underly the initial version of this draft, which need to be considered appropriately. Firstly, the role of CGAs in providing proof of address ownership in this protocol is primarily an enabler, and may be insufficient in some environments to authorize communications for a particular destination.

This may particularly be the case for devices with multiple users, where individuals have permission to access specific data services, but others are excluded. CGAs at host level granularity are insufficient to distinguish which of the users is attempting a communication.

Mechanisms which could be used to provide such distinction are user-level digital signatures over the filter message contents and freshness information, in addition to CGA information. Additionally, to ensure that the PDP only has to process a single digital signature, subsets users on multi-host systems could be allocated CGAs tied to their own public key information while using that host.

Therefore FCON messages from different users on a system would have different security credentials, and still allow CGA authorization. The internal processes on the host to allow for signing such messages is beyond the scope of this document, but it is easy to envisage an

ICMP message signing service which a user subscribes to, that uses the subscribed private-key credentials to sign FCON and SEND messages.

Regardless of the identity of the message signer, FCON request messages require authentication of the node, and the response messages require proof of the PDP's identity in order to prevent denial-of-service through spoofing attacks. Denial of service can occur due to the requirement to process digital signatures on response messages. Hosts which detect many excessive responses from PDPs, such as would indicate a denial-of-service attempt MAY defer processing digital signatures on responses, and rely on freshness and Nonce information in the message itself to determine if a digital signature needs to be processed. This limits denial . of service attacks to those who are able to guess nonces, or are on the path to the PDP.

16. Normative References

[I-D.ietf-nsis-nslp-natfw]

Stiemerling, M., Tschofenig, H., Aoun, C., and E. Davies,
"NAT/Firewall NSIS Signaling Layer Protocol (NSLP)",
[draft-ietf-nsis-nslp-natfw-18](#) (work in progress),
February 2008.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

- [RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 2434](#), October 1998.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", [RFC 2460](#), December 1998.
- [RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", [RFC 2710](#), October 1999.
- [RFC3024] Montenegro, G., "Reverse Tunneling for Mobile IP, revised", [RFC 3024](#), January 2001.
- [RFC3344] Perkins, C., "IP Mobility Support for IPv4", [RFC 3344](#), August 2002.
- [RFC3519] Levkowetz, H. and S. Vaarala, "Mobile IP Traversal of Network Address Translation (NAT) Devices", [RFC 3519](#), May 2003.
- [RFC3775] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", [RFC 3775](#), June 2004.
- [RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", [RFC 3810](#), June 2004.
- [RFC3963] Devarapalli, V., Wakikawa, R., Petrescu, A., and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol", [RFC 3963](#), January 2005.
- [RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", [RFC 3971](#), March 2005.
- [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", [RFC 3972](#), March 2005.

- [RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", [RFC 4213](#), October 2005.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing

Architecture", [RFC 4291](#), February 2006.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", [RFC 4861](#), September 2007.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", [RFC 4862](#), September 2007.

[Appendix A](#). Dynamic PDP Discovery (Informative)

In some of the described scenarios, it may not be feasible to prepopulate the endpoints with information for all of the relevant PDPs responsible for firewall policy along a particular data path. For these scenarios, sending data to an existing configured PDP may be insufficient to create state in all firewalls on the path. This may lead to packets being dropped, even though signalling has been performed, and all known signaling accomplished.

One mechanism to ensure that all PDPs for a data path are informed of changes is to send CREATE packets along the path at the time establishing a data connection. Such packets would be sent to the same IP source and destination as the packets in the data stream, but the IP packet would contain the Router-Alert Hop-By-Hop option [[RFC2460](#)].

Enforcement or Decision Point devices which receive such probes, would refer the packet for processing, and identify that the ICMP message contained a CREATE operation. The message filter contents would then define which sessions are requested to be allowed through the firewall.

Depending upon policy, the PDP or PEP would either DENY the data flow, sending a CREATE-NACK message, refer the sender to an appropriate PDP using the PDP-REDIRECT message, or create the state for the session according to the filter, and send a CREATE-ACK. In the case that the session state creation is refused or redirected, the CREATE packet itself is dropped. In the alternative case where state is created, the CREATE packet should be forwarded onward toward its destination. At this stage, the potential to create multiple responses to a single message is the primary danger of this method.

An alternative which increases setup latency is to drop the CREATE packet when new state is created, and send a CREATE-ACK. On each successive CREATE packet which does not alter session state, the CREATE packet is passed onward towards the destination. This removes any multiplication attacks, but causes delay of up to $N \times \text{RTT}$ for N policy devices along the path.

Similar operations for UPDATE would also occur, with state being created if it didn't exist, or replaced in the case it already was in place.

Once a host discovers that a Policy Decision Point exists on a particular path, it can then signal directly to it. Devices can add

these to the PDPs discovered using DHCP or other mechanisms.

Even though the packets are sent between the same source and destination addresses as those of the data session, they may travel a different path to the actual data stream, for example due to load balancing. In such a case, a PDP or PEP which receives the CREATE or UPDATE method can send a PDP-REDIRECT, to refer the origin to the appropriate policy decision point for the data flows.

Where a host application accepts inbound packets passively, it may be useful to ensure that Policy enforcement and decision points beyond the DHCP configured range are included in signaling. Such can be achieved by configuring a filter describing the allowed inbound source and destination addresses of packets and addressing a hop-by-hop CREATE packet to a set of potential inbound source addresses.

Such probing can be limited in terms of the number of addresses to probe, as well as the time-to-live of the packet itself, in order to prevent impacts upon the network.

Internet-Draft

Firewall Control Protocol

February 2008

Authors' Addresses

Hesham Soliman
Elevate Technologies

Email: hesham@elevatemobile.com

Greg Daley

Email: hoskuld@hotmail.com

Suresh Krishnan
Ericsson
8400 Blvd Decarie
Town of Mount Royal, Quebec
Canada

Email: suresh.krishnan@ericsson.com

Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be

found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).