Active Networks Group
Request for Comments: DRAFT
Category: Experimental

D. Scott Alexander
Bob Braden
Carl A. Gunter
Alden W. Jackson
Angelos D. Keromytis
Gary J. Minden
David Wetherall

Active Network Encapsulation Protocol (ANEP)

Status of this Memo

This memo defines an Experimental Protocol for the Internet community. This memo does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

Abstract

This document specifies a mechanism for encapsulating Active Network frames for transmission over different media. The suggested format allows use of an existing network infrastructure (such as IP [RFC791] or IPv6 [RFC1883]) or transmission over the link layer. In order to support ongoing research, the proposed mechanism is as generic and extensible as possible. This mechanism allows co-existence of different execution environments and proper demultiplexing of received packets.

1. Introduction

An active network node is capable of dynamically loading and executing programs, written in a variety of languages. These programs are carried in the payload of an active network frame. The program is executed by a receiving node in the environment specified by the ANEP. Various options can be specified in the ANEP header, such as authentication, confidentiality, or integrity.

This document describes the syntax and semantics of ANEP. The details of handling the contents of an active frame are left up to the individual implementations/environments.

2. Terminology

packet - an ANEP header plus the payload

active node - a network element that can evaluate active packets

TLV - acronym for Type/Length/Value constructs

basic header - the first two elements of the ANEP header

3. Raisons d'etre

The reasons an active network header is necessary are:

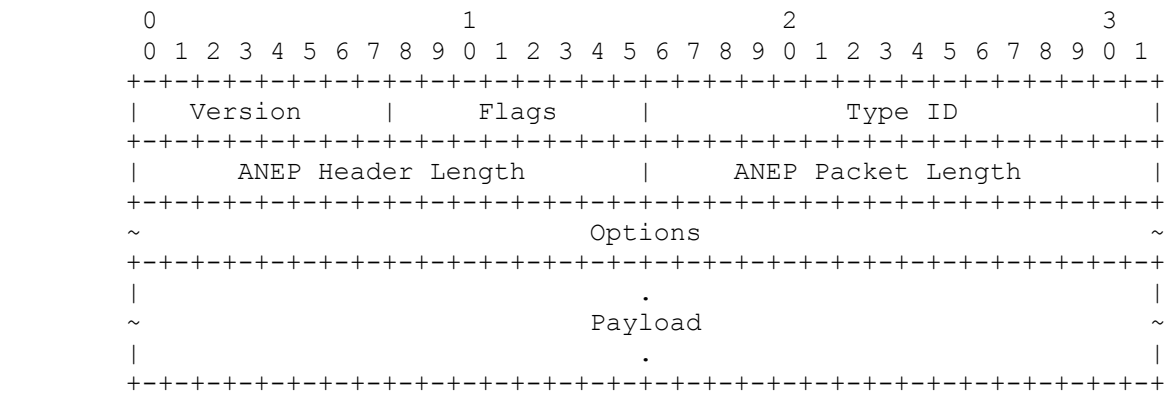
a) an active node receiving a packet must be able to uniquely and quickly determine the environment in which it is intended to be evaluated,

b) to allow minimal, default processing of packets for which the intended evaluation environment is unavailable, and

c) so that information that does not fit conceptually or pragmatically in the encapsulated program (such as security headers), can be placed in the header

4. Packet Format

The format of the ANEP header is:



All fields larger than one octet must be in network-byte order (big endian format). This holds for all Options as well.

The Version field indicates the header format in use. The version described by this document is 1. This field will be changed if the ANEP header should change. If an active node receives a packet whose version number it does not recognize, it should discard the packet. The length of this field is 8 bits.

The Flags field is 8 bits long. In version 1 of this protocol, only the most significant bit is used, to indicate what the node should do if it does not recognize the Type ID. If the value is 0, the node could try to forward the packet using the default routing mechanism (if one is in use), if the necessary information is available in the Options part of the header. If the value is 1, the node should discard the packet. The rest of the bits in this field should be ignored by the node. It is recommended that they be set to zero by the packet originator.

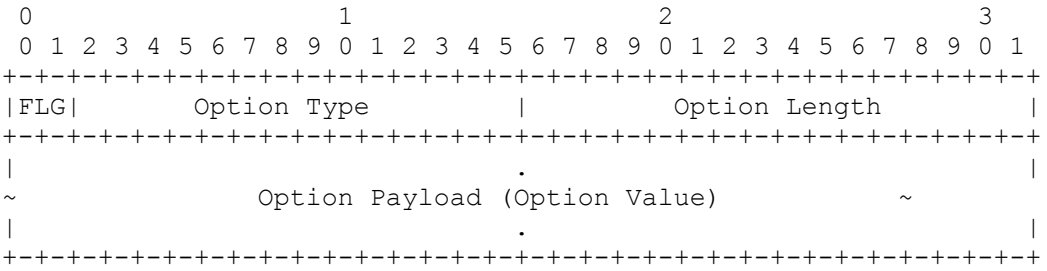
The ANEP Header Length field specifies the length of the ANEP header in 32 bit words. If no options are included in the packet, then its value must be 2. The length of this field is 16 bits.

The Type ID field indicates the evaluation environment of the message. The active node should evaluate the packet in the proper environment. The length of this field is 16 bits. The proper authority for assigning Type ID values to interested parties is the Active Networks Assigned Numbers Authority (ANANA). The Type ID value 0 is reserved for possible future network layer informational and error messages. If the value contained in this field is not recognized, the node should check the value of the most significant bit of the Flags field in deciding how to handle the packet.

The ANEP Packet Length field specifies the length of the entire packet, including the packet payload, in octets. This field is used to recover the packet if it has been transmitted over a lower layer that does not allow recovery of the packet length. The length of this field is 16 bits. Notice, that unlike other length fields in this document, the unit of measure is octets.

4.1 Options

Options in the form of TLVs can be included in the packet, immediately following the basic header. The format of these options follows:



The Option Type field identifies the option. How the active node handles the Option Payload depends on the Option Type value. The length of this field is 14 bits. The following values have been reserved:

Source Identifier	1
Destination Identifier	2
Integrity Checksum	3
N/N Authentication	4

All values intended for public use are under the authority of the Active Networks Assigned Numbers Authority (ANANA). Other parties can use their own values for this field if the most significant bit (Flags bit 0) is set. These Options are only meaningful inside the specified evaluation environment, so the proper authority for assigning these values is the Type ID owner.

The Option Length field contains the length of the TLV in 32 bit words. This includes the length of the Flags, Option Type, Option Length and Option Payload fields. This value must never be less than 1 (for an option with a zero sized Option payload). If the Option payload size is larger than the size of the data it carries, it is recommended that the excess 1 - 3 octets be zero filled and be ignored by a receiving implementation. The length of this field is 16 bits.

The 2 most significant bits of the first word in an Option (bits 0 and 1) are used as a Flag field. Bit 0 (Private) is used to indicate that the Option Type is only meaningful within the specified Type ID. The node should not try to parse the Option at packet receipt if this bit is set.

If the active node does not know how to process the indicated Option Type, the action taken is defined by the value of bit 1 of the Flags field.

0 - ignore this option and continue processing the header. It is recommended that the active node logs the event.

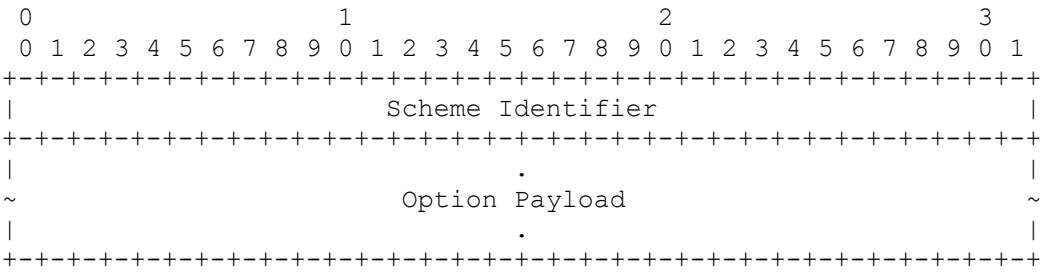
1 - discard the packet. It is recommended that the active node logs the event.

4.1.1 Defined Options

This section briefly discusses the options defined above.

4.1.1.1 Source Identifier

This Option includes a value which uniquely identifies the sender of the packet within the active network. The payload of this Option consists of a 32 bit value which identifies the addressing scheme in use, followed by that scheme's data. Notice that depending on the network infrastructure, the particular evaluation environment, and other options present in the packet (such as any authentication headers), this entry could be faked.



The following values have been reserved for the Scheme Identifier:

IPv4 address (32 bits)	1
IPv6 address (128 bits)	2
802.3 address (48 bits)	3

For 802.3 addresses, the remaining two bytes of the Option payload should be set to zero. IPv4 and IPv6 addresses are naturally aligned to 32 bits.

All other values are under the authority of the Active Networks Assigned Numbers Authority (ANANA).

4.1.1.2 Destination Identifier

This Option includes a value which uniquely identifiers an ultimate destination of the packet within the active network. The format of the payload of this Option is the same as that of the Source Identifier TLV (section 4.1.1.1). This field could be used by active nodes on which the intended evaluation environment is unavailable in order to attempt to forward the packet towards an active node capable of better

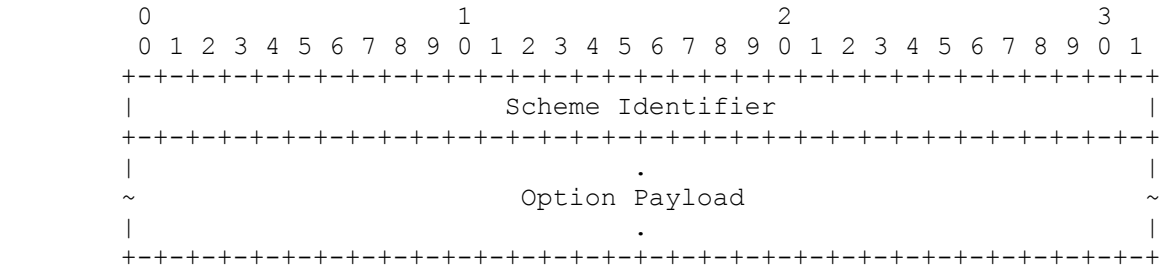
handling the packet.

4.1.1.3 Integrity Checksum

The payload of this Option contains the 16 bit one's complement of the one's complement sum of the entire ANEP packet, starting with the ANEP Version field [RFC1071] [RFC1141] [RFC1624]. For computing the checksum, the payload of this Option must be set to zero. The Option Length field must be 2.

4.1.1.4 Non-Negotiated Authentication

This option is used to provide one-way authentication, with no prior negotiation between the packet originator and processing node(s). The payload of this Option consists of a 32 bit value which identifies the authentication scheme in use, followed by that scheme's data. The Option Length field must never be less than 3.



The following values have been reserved for the Scheme Identifier:

SPKI Self-signed Certificate	1
X.509 Self-signed Certificate	2

All other values are under the authority of the Active Networks Assigned Numbers Authority (ANANA).

It is expected that this option will be used when the number of packets that require authentication is too small to justify the cost of a full negotiation, when the operation is time critical, or when security negotiation cannot take place. The processing cost of this option is expected to be higher than that of a negotiated authentication option, and it might not provide guarantees as hard as the latter, especially with respect to replay protection.

5. Security Considerations

No specific security mechanism has been specified in this document,

although an option number has been reserved. Additional Option numbers can be used to include more and/or different security services as necessary. Further research is required.

6. References

[RFC1883] Deering, S., R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC-1883, Internet Engineering Task Force, December 1995.

[RFC791] Postel, J.B., "Internet Protocol", RFC-791, September 1981.

[RFC1624] Rijsinghani, A., "Computation of the Internet Checksum via Incremental Update", May 1994.

[RFC1141] Mallory, T., Kullbert, A., "Incremental Updating of the Internet Checksum", January 1990.

[RFC1071] Braden, R., Borman, D., Patrtidge, C., "Computing the Internet checksum", September 1988.

Appendix A: Processing of Options

Options are first examined on packet receipt, in their order of appearance in the packet. Options that have their Private bit set should be ignored by the implementation at this stage. All Options defined in this document require no action from the active node.

Information on the Options may also be made available to the packet program during its evaluation. Information on the Options may be used by the evaluation environment directly.

There is no priority for dealing with Option parsing failure. As soon as a node tries to parse an unknown Option, it takes action depending on the Flags field of the Option. If the node examines the Options List in order to find the necessary information to forward the packet, it should not try to parse unknown Options nor raise an exception when one is encountered.

Appendix B: Active Networks Assigned Numbers Authority

Currently (as of July 17, 1997) Bob Braden (braden@isi.edu) is acting as the ANANA. Should this change in the future, a draft will be issued. For contact information, see the Authors' Addresses section, later in this draft.

Authors' Addresses:

D. Scott Alexander
Distributed Systems Lab
Computer and Information Science Department
University of Pennsylvania
200 South 33rd Street
Philadelphia, PA 19104 - 6389
Email: salex@dsl.cis.upenn.edu

Bob Braden
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
Phone: (310) 822-1511
Email: Braden@ISI.EDU

Carl A. Gunter
Computer and Information Science Department
University of Pennsylvania
200 South 33rd Street
Philadelphia, PA 19104 - 6389
Email: gunter@saul.cis.upenn.edu

Alden W. Jackson
BBN Technologies
10 Moulton Street
Cambridge, MA 02138
Email: awjacks@bbn.com
Phone: +1 (617) 873-3000

Angelos D. Keromytis
Distributed Systems Lab
Computer and Information Science Department
University of Pennsylvania
200 South 33rd Street
Philadelphia, PA 19104 - 6389
Email: angelos@dsl.cis.upenn.edu

Gary J. Minden
The University of Kansas
Information and Telecommunications Technology Center
2291 Irving Hill Road
Lawrence, KS 66045
Email: gminden@ittc.ukans.edu
Phone: 785.864.4834
Fax: 785.864.7789

David Wetherall
MIT Lab. for Computer Science
545 Technology Sq. (room 504)
Cambridge MA 02139
Email: djw@lcs.mit.edu