Network Access Control Protocol (NACP)

Intellectual Property Statement

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.

Status of this Memo

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/1id-abstracts.txt
The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Abstract

The Network Access Control Protocol (NACP) is a protocol used to encapsulate EAP (Extensible Authentication Protocol) messages in a UDP (User Datagram Protocol) transport between an authenticator and peer.

Conventions used in this document
1. Introduction

EAP is a protocol used to communicate authentication and other information between an authenticator and peer for the purposes of enforcing access control at a particular device in the network.

EAP has been defined to run over multiple L2 transports including wired and wireless LANs (802.1x) and PPP (PPP-EAP).

This document specifies the Network Access Control Protocol (NACP), a protocol that encapsulates EAP messages in a UDP transport between an authenticator and peer. The authenticator and peer may be one or more L3 hops away from each other.

1.1 Terminology
peer: End-system device that responds to EAP messages from authenticator

authenticator: Network access device that initiates the EAP exchange

association: A NACP connection between authenticator and peer.

2. Protocol Overview

2.1 Messages

There are four NACP messages:

Hello message: Used to set up an association between authenticator and peer

Validate message: Used to transport EAP payloads between an authenticator and peer, with exception of EAP Success and EAP Failure payloads

Result message: Used to transport EAP Success and EAP failure payloads between authenticator and peer

NAK message: Used for NACP version negotiation

2.2 Message Contents

2.2.1 Message Header

The NACP header contains the following fields (see Section 4 for the message format details):

- Version of protocol in use. This specification of the protocol is version 1.
- Message operation code (opCode)i.e. Hello, Validate, Result or NAK message
- Flags indicating whether this message is a request or response
- Length of payload
- Message identifier (Message ID) for use in sequencing and retransmissions
- Association identifier (Association ID) for uniquely identifying an NACP connection between a peer and authenticator.

2.2.2 Message Payload
The following Type-Length-Value (TLV) tuples may be present in the NACP payload depending on the message type:

Association ID TLV. Communicates value of initial association ID between authenticator and peer

EAP Payload TLV. Used to encapsulate EAP messages between authenticator and peer

Cookie TLV. Used to protect peer from a blind resource consumption attack.

2.3 Association

The authenticator establishes a NACP association for the purposes of authenticating and authorizing a peer using the EAP protocol. At a minimum, an association must be maintained on the authenticator and peer for the duration of the Hello-Validate-Result sequence to ensure that the EAP message exchange completes. An association should be maintained for a longer period of time where it is expected that the association will be used for other purposes e.g. periodic EAP re-authentication purposes. In such cases, the association is terminated based on an idle time-out or other event on the peer and authenticator.

An association is uniquely identified on the authenticator by the peer’s IP address and an identifier that is chosen to be unique on the authenticator. Similarly, an association is uniquely identified on the peer by the authenticator’s IP address and an identifier that is chosen to be unique on the peer. The association identifier used on the peer and authenticator MUST be a random number greater than zero to protect against blind attacks and delayed packets.

When the authenticator sends a message, it uses the peer’s association identifier in the Association ID field; when the peer sends a message it uses the authenticator’s association identifier in the Association ID field. The initial values of the Association IDs are exchanged in the Association ID TLV in the Hello payload.

2.4 Sequencing and Retransmission

NACP is a request-response protocol. In any particular message exchange, one party acts as the initiator (sends a NACP request) and the other party acts as the responder (sends a NACP response message). In version 1 of the NACP protocol, the authenticator is always the initiator and the peer always the responder.
The initiator sets the Message ID in the header to any value (such as zero) in the first message of the NACP association, and increases the Message ID by one for each new request using serial number arithmetic. Retransmissions do not increment the Message ID. The responder sets the message ID in the response to the value of the message ID in the request.

The initiator is always responsible for retransmissions. The responder only retransmits a response on seeing a retransmitted request; it does not otherwise process the retransmitted request. The retransmitted requests/responses are exact duplicates of previous requests/responses.

The initiator must not send a new request until it receives a response to the previous one. Packets with out-of-sequence Message IDs are considered invalid packets and are discarded.

The initiator must retransmit after a configurable interval until either it gets a valid response, or decides after a configurable number of attempts that the NACP association has failed. (Since the retransmission algorithm is implementation-dependent, it is not defined here.)

2.5 Message Validity Check and Error Handling Rules

A message is only accepted if all the following holds true:

- Message version field = 1.
- Association ID must match a current association
- All messages received by peer have R bit in flag set to zero
- All responses received by authenticator have R bit in flag set to one.
- Message opCode is valid
- Message length equals size of payload
- Message ID must match the expected sequence number
- The payload contains only those TLVs expected given the value of the opCode (see Section 4)
- All TLVs within the payload are well-formed. Also, TLVs marked as mandatory are recognized.

Processing of an invalid message is as follows:

2.5.1 Invalid Version

If a responder receives a request with a message version that is higher than the version it supports, the responder MUST send a response with an opcode set to NAK, and with the version number field set to the highest version it can support. The responder does not further process the received packet.
If the responder sends a version different from the version sent by the initiator, the packet will be silently discarded, if the opcode is not set to NAK.

2.5.2 Invalid Association

The Association ID must match a current association, except in the first Hello request from the authenticator to the peer where the Association ID is set to zero.

If the Association ID is invalid, the receiver silently discards the packet.

2.5.3 Message ID

If the Message ID does not conform to the rules in the sequencing and retransmission section, the packet is discarded.

2.5.4 Cookie TLV

The cookie TLV is sent by the peer in a Hello response. The authenticator includes the cookie TLV in the first Validate message of the NACP association. If the cookie sent by the peer is not the one received, the peer discards the packet.

2.5.5 Other

If any of the other conditions are not met, the packet is also discarded.

2.6 Fragmentation

NACP does not provide support for fragmentation. The EAP method may support fragmentation and/or the IP layer may be used to support fragmentation where the link-layer MTU requires it.

2.7 Transport Protocol

NACP uses UDP as the transport protocol with port number TBD. All messages are unicast.

3. Protocol Operation

3.1 Hello Message

PEER                                             AUTHENTICATOR
When a new NACP association is initiated, the authenticator sends a Hello message to the peer.

The Hello message contains the Association ID TLV. The authenticator chooses a random non-zero 32-bit number for the Association ID that is unique across all current associations.

The peer responds to the Hello message with an Association ID TLV of its own in the payload. The peer chooses a random 32-bit number to uniquely identify its end of the association and sends it in the Association ID TLV in the Hello payload. The peer also sends a cookie in the Cookie TLV. The authenticator must include this TLV in the subsequent request message so that the peer can verify that the authenticator is the one that initiated the NACP conversation. The peer does not maintain state about the association with the authenticator until the next request message is received with a valid Cookie TLV.

### 3.2 Validate Message

<table>
<thead>
<tr>
<th>PEER</th>
<th>AUTHENTICATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;-Validate(MsgID=x+1,AssocnID=M)[EAP Payload][Cookie TLV]-----------</td>
<td>--Validate(MsgID=x+1,AssocnID=N)[EAP Payload]-------</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>&lt;-Validate(MsgID=x+y,AssocnID=M)[EAP Payload]----------------------</td>
<td>--Validate(MsgID=x+y,AssocnID=N)[EAP Payload]-------</td>
</tr>
</tbody>
</table>

Once an association is set up using the Hello exchange, the authenticator uses EAP to request credentials from the peer as defined in [EAP]. The authenticator encapsulates EAP request and response messages (but not Success and Failure messages) in Validate packets.

The authenticator includes the Cookie TLV that it received in the Hello response in the first Validate request. The authenticator does not otherwise process the cookie. Authenticator does not include the Cookie TLV in any of the subsequent Validate messages.

The peer responds to a Validate request following normal processing rules. If this is the first Validate request in the NACP association, the peer also validates the cookie before creating association state.
The Validate message exchange lasts for as long as the EAP Layer requires, and typically ends in the sending of an EAP Success/Failure using the Result message described below.

### 3.3 Result Message

<table>
<thead>
<tr>
<th>PEER</th>
<th>AUTHENTICATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;-Result(MsgID=x+y+1,AssocnID=M)[EAP Success/Failure] ----</td>
<td>--Result(MsgID=x+y+1,AssocnID=N)------------------------&gt;</td>
</tr>
</tbody>
</table>

The authenticator encapsulates an EAP Success or Failure message in the NACP Result message. Note that where the authenticator is acting as EAP pass-through, it needs to look into the EAP header to determine when an EAP Success/Reject is being sent.

The peer responds to a Result request with a Result response. A Result response carries no payload.

On receiving a response to the Result message, the authenticator validates the Result packet according to NACP processing rules described above. No further processing is done since the response is empty.

### 3.4 NAK message

If the peer receives a request with a message version that is higher than the version it supports, the peer MUST send a response with an opcode set to NAK, and with the version number field set to the highest version it can support. The peer does not further process the packet.

### 3.5 Example Operation

When an association is established, the following flow takes place:

<table>
<thead>
<tr>
<th>PEER</th>
<th>AUTHENTICATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;-Hello(MsgID=x,AssocnID=0)[AssocnID=N]----------</td>
<td>--Hello(MsgID=x,AssocnID=N)[AssocnID=M][Cookie-TLV]------</td>
</tr>
<tr>
<td>&lt;-Validate(MsgID=x+1,AssocnID=M)[EAP Identity][Cookie TLV]-----</td>
<td>--Validate(MsgID=x+1,AssocnID=N)[EAP Identity]---------</td>
</tr>
<tr>
<td>&lt;-Validate(MsgID=x+2,AssocnID=M)[EAP method]--------</td>
<td>--Validate(MsgID=x+2,AssocnID=N)[EAP method]----------</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
</tr>
</tbody>
</table>
Once an association exists, the following flow takes place e.g. on EAP re-authentication:

\[
\begin{array}{c}
\text{PEER} \\
<-\text{Validate (MsgID=x+y, AssocnID=M) [EAP method]} \rightarrow \\
--\text{Validate (MsgID=x+y, AssocnID=N) [EAP method]} \rightarrow \\
<-\text{Result (MsgID=x+y+1, AssocnID=M) [EAP Success]} \rightarrow \\
--\text{Result (MsgID=x+y+1, AssocnID=N)} \rightarrow \\
\end{array}
\]

4. Message Format

All messages are transmitted in network byte order.

The message format has the following structure:

\[
\begin{array}{c}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 0 & 1 \\
\end{array}
\]

\[
\begin{array}{c}
+---------------------------------+ \\
| Message Header \\
+---------------------------------+ \\
| Message Payload \\
/ (Zero or more TLVs) \\
\end{array}
\]

The subsections below describes the format of the header and payload.

4.1 Header Format
### Flags
- 8-bit field indicating options
- `0 1 2 3 4 5 6 7`
- `M` | Reserved
- `--------`

**Response Flag (R).**
- Set to 1 when message is a response to a request (same message ID). R must be zero in all request messages.

**Reserved Flags:**
- Reserved. Must be zero. MUST be cleared on sending and ignored on receipt.

**Version:**
- 4-bit field indicating the version of this protocol. This specification = 1

**Opcode**
- 4 bit field indicating NACP message type
  - NAK = 1 (response only)
  - Hello= 2
  - Validate = 3
  - Result = 4

**Length**
- Length of NACP payload in octets excluding header

**Message ID**
- Used to ensure ordered delivery and detect retransmissions.

**Association ID**
- Uniquely identifies association in authenticator and peer.

### 4.2 Payload Format
The payload consists of zero or more type-length-value 3-tuples (TLVs).

Note: TLVs may not always start on a 4-byte boundary.

4.2.1 General TLV Format

The general TLV format is shown below.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Flags         |     Type      |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Value ...                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Flags

8-bit field indicating options

```
0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|M| Reserved    |
+-+-+-+-+-+-+-+-+
```

Mandatory Flag:
When set to one, the TLV Type is mandatory to support. If the Type is not recognized, the packet is discarded. When the M flag is set to zero, the TLV is optional. If an optional TLV Type is not recognized, it is ignored and processing of the packet continues.

Reserved:
Must be zero. Cleared on sending and ignored on receipt.

Type

8-bit field indicating type of Value field.

1= Association ID TLV
2 = EAP Payload TLV
3 = Cookie TLV

Length

16-bit field indicating length of Value field in octets

Value

Value field depends on Type.

4.2.2 Association ID TLV
Network Access Control Protocol (NACP)  May 2005

0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Flags         |     Type      |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Association ID                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Flags
   Mandatory Flag: Set to 1

Type
   1

Length
   4

Association ID
   Locally selected non-zero 32-bit random number that uniquely
   identifies an association on authenticator and peer.

The authenticator uses this TLV in a Hello request to indicate to the
peer what association ID to use in all packets the peer sends to the
authenticator (including the Hello response). The peer uses this TLV
in a Hello response to indicate to the authenticator what association
ID to use in subsequent packets the authenticator sends to the peer.

4.2.3 EAP Payload TLV

0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Flags         |     Type      |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         EAP Payload                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Flags:
   Mandatory Flag: Set to 1

Type:
   2

Length
   Length of EAP Payload in octets

EAP Payload
   EAP Packet
An EAP Payload TLV is used in Validate messages to carry EAP Requests and Responses, and in the Result message (Request only) to send an EAP Success or Failure message.

### 4.2.4 Cookie TLV

<table>
<thead>
<tr>
<th>Flags</th>
<th>Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

#### Flags
- Mandatory Flag: Set to 1

#### Type
- 3

#### Length
- Length of Cookie in octets

#### Cookie
- Cookie algorithm is implementation-dependent. It consists of information sufficient to establish the association with the authenticator after the first Validate request us received and is protected by a secret known only to the peer.

The Cookie TLV is used by the peer to protect itself against a DOS attack from a device that is blindly launching spurious Hello requests. The peer must include the cookie TLV in a Hello response. The authenticator must return the cookie TLV in the first Validate request of the association only. The peer does not establish state until the cookie in the first Validate message has been received and validated. The cookie algorithm is not defined in this specification as it is not needed for interoperability.

### 4.2.5 Summary

This section summarizes the TLVs that are valid for each message.

- NAK Response: None
- Hello Request: Association ID TLV
- Hello Response: Association ID TLV, Cookie TLV
Validate Request: EAP Payload TLV, Cookie TLV (first Validate only)
Validate Response: EAP Payload TLV
Result Request: EAP Payload TLV
Result Response: None

5. Security Considerations

We consider NACP security vulnerabilities in this section.

5.1 Blind Denial of Service Attacks

An attacker may carry out a number of blind denial of service attacks on a peer or authenticator using NACP:

1. An attacker may attempt to mount a denial of service attack on a peer or authenticator by injecting NACP packets into an existing association
2. An attacker may attempt to mount a denial of service attack on the peer by spoofing Hello messages

The Association ID is a random number and can be used to protect an association from being compromised by a blind packet injection attack.

The Cookie TLV protects a peer against fraudulent authenticators spoofing Hello messages with the intent of consuming resources on the peer and denying legitimate authenticators from communicating with the peer.

5.2 On-path Attacks

An attacker that has access to NACP packets may eavesdrop on an association, or launch an attack on an association by inserting, deleting, modifying and replaying packets causing an authenticator or peer to discard or misinterpret packets and compromise the association.

NACP does not provide protection against these forms of attacks. In particular, NACP does not protect the EAP message exchange where eavesdropping and man-in-the-middle attacks are a significant risk. It is therefore the responsibility of the network operators and end users to choose the proper EAP method to protect their particular environment. The contents of EAP messages transported inside NACP must be protected using an EAP method that meets the requirements for securing the transfer of authentication and other information for the purposes of network access control. Security considerations for EAP methods are discussed in [EAP].
If breach of confidentiality and deliberate attacks on the integrity of the NACP protocol itself are a significant risk in certain deployment environments, NACP should be protected by a protocol that offers confidentiality and/or packet authentication, integrity and protection against replay e.g. IPSEC [IPSEC].

6. IANA Considerations

This section provides guidance to the Internet Assigned Numbers Authority (IANA) regarding registration of values related to NACP, in accordance with BCP 26, [RFC2434].

NACP uses a UDP port number that needs to be assigned by IANA.

The following namespaces in NACP require registration: OpCodes, TLV Types.

7. References

7.1 Normative References


7.2 Informative References


Acknowledgments

We acknowledge the contributions of Saul Adler, Eli Gelasco and Dany Rochefort.

Authors’ Addresses

Joseph Salowey
Cisco Systems
2901 3rd Ave
Seattle, WA 98121
USA
Email: jsalowey@cisco.com

Susan Thomson
Cisco Systems
499 Thornall St, floor 8
Edison, NJ 08837
USA
Email: sethomso@cisco.com

Fan Wu
Cisco Systems
5400 Airport Boulevard
Boulder, CO 80301
USA
Email: wufan@cisco.com

Venkateswara Rao Yarlagadda
Cisco Systems India (Pvt) Ltd.
No. 9 Brunton Road
Bangalore, Karnataka
India
Email: yvrao@cisco.com

Hao Zhou
Cisco Systems
4125 Highlander Parkway
Richfield, OH 44286
USA
Email: hzhou@cisco.com
Full Copyright Statement

Copyright (C) The Internet Society (2005).

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 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.