Internet-Draft                     Yoshihiro Ohba/Shinichi Baba
Expires: March, 2003            Toshiba America Research, Inc.
Subir Das
Telcordia Technologies
September 30, 2002

PANA over TLS

<draft-ohba-pana-potls-00.txt>

Status of This Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC 2026.

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

This draft specifies a method to carry authentication information over TLS between PANA Client (PaC) and PANA Authentication Agent (PAA). PANA over TLS uses existing TLS protocol over a reliable transport in order to perform authentication information exchange in a secure and reliable manner. The purpose of this document is not only to provide a mechanism for carrying the authentication parameters but also to address some outstanding issues such as, multiple access routers, reauthentication, security threats, etc.
1 Introduction

This protocol, PANA over TLS (Protocol for carrying Authentication for Network Access over Transport Layer Security), is designed for authentication message exchange between PaC and PAA, both of which are on the same subnet [PANAREQ].

PANA over TLS uses TLS [TLS] over a reliable transport in order to perform authentication information exchange in a secure and reliable manner. In particular, the security features provided with TLS are important for providing encryption and/or integrity protection for the entire authentication protocol exchange including the identity of the client as well as authentication result (e.g., EAP-Success/Failure) that is not protected in some authentication protocol such as EAP [EAP]. Without protecting those information it is difficult to distinguish the case where authentication is failed.
due to invalid credentials from other errors that might have happened as a result of some active attack.

There are a number of protocols such as IKE (Internet Key Exchange) [IKE] and PIC (Pre-IKE Credential provisioning) [PIC] that could be used for protecting authentication message exchange over a secure communication channel. However, TLS is selected in this protocol for the following reasons.

First, unlike IKE, TLS does not require mutual authentication for establishing a secure communication channel between peer entities. It would not be a realistic requirement for assuming mutual authentication especially in roaming environments. Second, unlike PIC, TLS supports a session resumption functionality that can be used for making re-authentication faster than that is performed without session resumption.

PANA over TLS is designed for carrying any authentication protocol information including EAP messages. It is also possible to use a TLS certificate for authenticating a PaC without using any other authentication protocol. PANA over TLS supports combining multiple types of authentication to authenticate a PaC. For example, it is possible to use a TLS client certificate for authenticating an IP address of the PaC and then use EAP for authenticating the user of the PaC.

PANA over TLS also defines a Device Identifier [PANAREQ] protection mechanism to protect man-in-the-middle attacks against transport connections by spoofing MAC/IP address while man-in-the-middle attacks against contents carried over TLS connections are protected by TLS.

2. Requirements Language

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

3. Protocol Overview

In this protocol authentication information is carried between a PaC and a PAA over a secure TLS connection on top of a reliable transport protocol such as TCP and SCTP. The PaC and the PAA are the client and the server of the TLS connection, respectively. The TLS connection is established based on the TLS Handshake Protocol defined in [TLS].

When authentication information is carried over a TLS connection, confidentiality and integrity for the authentication information
exchange between the PaC and PAA are provided by the TLS connection. Reliability, congestion control and fragmentation free communication are provided by the reliable transport (though TLS also handles fragmentation).

To establish a TLS connection a PaC needs to find (an) IP address(es) of the PAA(s) on the link. PAA Discovery described in this document is used for this purpose.

It is possible to use a TLS client certificate that is optionally carried during a TLS handshake as the credential of PaC. In this case, additional authentication MAY be performed over the TLS connection.

In order to avoid Device Identifier spoofing, PANA over TLS also supports exchanging Device Identifier information over the TLS connection.

PANA over TLS supports fast authentication based on the session resumption functionality of TLS [TLS].

The security association corresponding to the master secret of the TLS session established between PaC and PAA is considered to be a Local Security Association (LSA) from which other security association can be derived. The TLS master secret can be used for deriving any kind of security association.

PANA over TLS is designed to work over both multi-access links and point-to-point links (this does not necessary mean PPP links, an IP-in-IP tunnel and a GRE tunnel are also point-to-point). The only requirement is the PaC and PAA is on the same link in order to discover PAAs based on link-local multicast.

4. Protocol Specification

4.1. TLS Session, TLS Connection and Transport Connection

A TLS connection is a secure data channel over which application data is securely exchanged between TLS peers. Most of the messages defined in this protocol are carried over TLS connections.

A TLS session is a signaling channel used for establishing a master secret shared between TLS peers and establishing a TLS connection by negotiating cipher suites based on the master secret. The lifetime of a TLS session can be longer than that of a TLS connection, i.e., after terminating a TLS connection a new TLS connection can be established or an existing suspended TLS connection can be reused.
within the same TLS session (TLS session resumption). As specified in [TLS], TLS Alert/close_notify messages need to be exchanged before terminating the TLS connection in order for a session to be able to be resumed later.

A transport connection that is associated with a TLS connection MUST be terminated when the current TLS connection is terminated. Similarly, the current TLS connection that is associated with a transport connection MUST be terminated when the transport connection is terminated. After successful authentication, the transport connection MUST be kept open as long as the PaC is authorized for network access when Authenticated Heartbeat Protocol (AHP) is used for detecting implicit disconnection of a PaC. Otherwise, the transport connection MAY be terminated.

4.2. Transport Layer Protocol

PANA over TLS uses TCP or SCTP as the TLS transport. SCTP is preferable since it has a cookie-based 4-way handshake mechanism to protect against masquerade attacks (e.g., TCP SYN attacks) for transport connections.

UDP is also used for carrying messages used for finding (an) IP address(es) of PAA(s) and requesting (re-)authentication from PAA(s).

Both PaC and PAA need to configure an IP address before running this protocol. In IPv6, a link-local address can be used for this protocol. In IPv4, the IP address that is currently assigned to the interface is used.

The same port number PortNumber is used for TCP, SCTP and UDP.

4.3. PAA Discovery

/* Authors’ note:

For ease of understanding and completeness of this document the following section has been described here. However, the authors recognize that PAA discovery is a problem by itself and need further discussions.

*/

Assuming that PAA is not co-located with an access router, a discovery mechanism is necessary for determining an IP address of the PAA. In addition, there may be multiple enforcement points and all of them may or may not be controlled by a single PAA. Methods that are used for a PaC to choose one or more PAA(s) when there are multiple PAA(s) in the same subnet are for further study. In any situation, a
PAC needs to be aware of at least an IP address of each PAA and such information can be obtained by using at least one of the following ways:

1. Manual configuration
2. Notification from PAA
3. DHCP
4. Multicast query

Each method is explained below.

4.3.1. Manual Configuration

This is entirely specific to implementation and not described in this document.

4.3.2. Notification from PAA

When a PAA detects that a new PaC device is connected to the subnet, it MAY send an AuthRequest message to the PaC. The AuthRequest message is unicast over UDP. The PaC that receives an AuthRequest message will start establishing a TLS connection with the PAA. The PaC SHOULD NOT start establishing a TLS connection when it receives an from a PAA if it is in mid of performing authentication with the PAA. How a PAA detects the presence of a new PaC is out of the scope of this document.

4.3.3. DHCP

A new DHCP configuration option needs to be defined to carry the information described above.

4.3.4. Multicast Query

When PAA Discovery is performed via multicast, a PaC sends a PAADiscover message over a specific link-local multicast address "All-PAA-Nodes." Each PAA that received the message responds with an AuthRequest message. The AuthRequest message is unicast over UDP. A PAA SHOULD silently discard a PAADiscovery message received from a PaC without responding with an AuthRequest if it is mid of performing authentication with the PaC.

In the case of IPv4, All-PAA-Nodes is the same as "all-hosts" group
In the case of IPv6, All-PAA-Nodes is a link-local scoped multicast address to be assigned by IANA.

All PAAas MUST support this method.

4.4. Authentication Modes

There are two modes of authentication: Full authentication and Fast Authentication. Full Authentication is performed when a new TLS session is established with a full TLS handshake. A new session ID is allocated for the session. For Full Authentication, a server certificate MUST be used in TLS handshake to avoid a man-in-the-middle attack where TLS connections are spliced at an intermediate eavesdropper.

Fast Authentication is performed based on an existing TLS session by using the session resumption functionality of TLS. A PaC can always propose Fast Authentication whenever it has a TLS session with the PAA. Fast Authentication is performed by specifying the session ID of the existing TLS session. On the other hand, The PAA can always choose to either perform Fast Authentication or force Full Authentication for the PaC that is proposing Fast Authentication with an existing session ID.

4.5. Authentication Types

There are two types of authentication defined for Full Authentication mode: Non-TLS Authentication and TLS Authentication. In Non-TLS Authentication, a PaC is authenticated without using a TLS client certificate. In Non-TLS Authentication, a PAA MUST NOT request a TLS client certificate during a TLS handshake, and AuthInfo message exchange MUST be performed over the TLS connection. In TLS Authentication, a TLS client certificate MUST be used during TLS handshake for authenticating a PaC and additional AuthInfo message exchange MAY be performed over the TLS connection. There is no distinction in authentication type for Fast Authentication.

4.6. Multi-level Authentication

PANA over TLS supports multi-level authentication in which multiple legs of "sub-authentication" may be performed one by one until a PaC is finally authenticated by a PAA. For example, in the case of TLS authentication with AuthInfo message exchange, authentication during TLS session negotiation with a TLS client certificate can be a sub-authentication leg and authentication based on AuthInfo message exchange can be another sub-authentication leg. Or when EAP message
is carried in AuthInfo message, multi-level authentication can be
performed within EAP [EAPBIS].

When multi-level authentication occurs, it is the matter of
authorization policy whether the entire sub-authentication legs need
to be successful in order for a PaC to be finally authenticated or a
restricted level of authorization may be applied when only a portion
of the entire sub-authentication legs is successful, except that TLS
authentication always requires successful TLS client certificate
authentication to establish a TLS session. Such an authorization
policy issue is out of the scope of this document.

4.7. Protecting Device Identifiers

Although TLS server certificates used for TLS handshake prevents man-
in-the-middle attacks against TLS connections, another type of man-
in-the-middle attack is still possible against transport connections.
That is, an intermediate attacker may splice two transport
connections that carry the contents of a single TLS connection
between a PaC and a PAA without any modification. As a result, the
attacker can successfully make its own IP address authorized for
network access instead of the IP address of the PaC. Also, if a TLS
server certificate is not directly associated with an IP address of
the PAA, it is also possible for the intermediate attacker to be a
rogue PAA (the TLS server certificate may be associated with a FQDN,
but the FQDN may result in being mapped to a different IP address if
DNS is not secured).

To deal with the man-in-the-middle attack against transport
connections, PANA over TLS defines a Device Identifier exchange
mechanism over TLS connections. DeviceID message is used for this
purpose.

If a received Device Identifier contained in a DeviceID message sent
from the peer is different from that is actually specified in the IP
and/or MAC header(s) of the underlying transport connection, the PAA
MUST return an Error message and immediately terminate the TLS
connection and transport connection.

4.8. Authenticated Heartbeat Protocol

PANA over TLS defines Authenticated Heartbeat Protocol (AHP) in which
short messages (i.e., Heartbeat) are exchanged over the TLS
connections in order to detect an inactive PaC without allowing an
attacker to be able to gain authorized network access by spoofing the
IP address of the legitimate PaC. Both Fast Authentication and AHP
can be used for local re-authentication in which re-authentication is
performed locally between PaC and PAA based on an established local security association (i.e., the TLS session) between them [PANAUSAGE]. Re-authentication can be performed faster by AHP than by Fast Authentication at the cost of holding the transport connection. AHP MUST be used if the layer 2 entity does not provide device disconnect indication to higher layer entities.

A PAA can send a Heartbeat/Request message whenever it needs to check whether the PaC is active or not. The PaC is requested to send a Heartbeat/Response message in response to the Heartbeat/Request message. Heartbeat/Request message MAY be sent periodically.

AHP starts after successful Full or Fast Authentication. When AHP is used, the transport connection MUST remain established. The PaC device is considered to be inactive when no Heartbeat message is received within a Heartbeat/Response timeout period, and an appropriate action SHOULD be taken by the PAA for the device, e.g., deleting the authorized status at the enforcement point(s) controlled by the PAA.

DefaultHeartbeatTimeout is the default timeout value for Heartbeat/Response.

A PaC that is running AHP may determine to disconnect from the network. If this happens, it MAY terminate the TLS connection and the transport connection without terminating the TLS session for future Fast Authentication based on session resumption. This is achieved by exchanging TLS Alert/close_notify messages between peers before terminating the TLS connection and the transport connection.

4.9. Deriving Purpose-Specific Keys

The master secret of an established TLS session can be used for deriving a cryptographic key that can be used for a specific purpose. This purpose-specific key can be used by some other protocols for their client/server authentication. This document describes a generic rule for deriving such a purpose-specific key.

\[
purpose\_specific\_key = PRF(SecurityParameters.master\_secret, \\
PURPOSE\_SPECIFIC\_STRING, \\
SecurityParameters.server\_random + \\
SecurityParameters.client\_random).
\]

The definition of the PRF function and the structure of SecurityParameters are specified in [TLS].
rules depend on each purpose, and thus are not defined in this document.

4.10. Message Flows

Message flows for possible combination of authentication types and modes are illustrated in Figures 1 through 3. In these figures, messages marked with * are defined in this protocol and carried over the secure TLS connection as application data. Messages not marked with * are TLS signaling messages defined in [TLS], except for PAADiscover and AuthRequest messages that are carried over UDP. Messages surrounded in a pair of square brackets are optional. Although this protocol can carry any authentication protocol information, EAP messages are typically carried in AuthInfo messages.

```
PaC                                                  PAA
[PAADiscover]                  -------->                  [AuthRequest]
                        <--------    
ClientHello                   -------->                  ServerHello
                        Certificate
                        ServerKeyExchange
                        <--------                  ServerHelloDone
ClientKeyExchange               Certificate
CertificateVerify
ChangeCipherSpec
Finished                      -------->                  ChangeCipherSpec
                        <--------                  Finished
DeviceID*                       <--------                  DeviceID*
AuthInfo*                       <--------                  AuthInfo*
                        .
                        .
                        <--------                  Success/Failure*
[Heartbeat*]                   <--------                  [Heartbeat*]
                        .
                        .
```

Figure 1: Message Flow for Full Non-TLS Authentication

Expires March, 2003
4.11. Message Formats and Processing Rules

In this specification, all multi-octet fields are encoded in network byte order.

All messages defined in this document start with the following header.
4.11.1. PAADiscover Message

When this message will be sent:

This message is multicast over UDP by a PaC needs to know (an) IP address(es) of PAA(s) to perform Full Authentication.

Meaning of this message:

This message means that the sender PaC is searching PAA(s). When a PAA receives this message from a PaC, it SHOULD return a AuthRequest message to the PaC unless if it is in mid of authentication with the PaC.

Structure of this message:

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

Type

A type of the message.

Subtype

A type-specific information needed for decoding the message payload. The Subtype name "NoSubtype" indicates that there is no subtype for that Type.

Length

An unsigned 2-octet integer that contains the length of the message in octets, including the header and payload of the message.

```
0x01  PAADiscover
```
Subtype

Subtype value     Subtype name
0x00              NoSubtype

4.11.2. AuthRequest Message

When this message will be sent:

This message is sent over UDP by a PAA when Full or Fast Authentication is needed.

Meaning of this message:

When a PaC receives this message, it starts Full or Fast Authentication if the PaC device needs to be authorized for network access. The PaC SHOULD NOT start establishing a TLS connection when it receives an from a PAA if it is in mid of performing authentication with the PAA.

Structure of this message:

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Subtype    |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
Type

Type value     Type name
0x02        AuthRequest

Subtype

Subtype value     Subtype name
0x00                   NoSubtype

Length

The Length value is 4.

Payload

The payload is null.

4.11.3. DeviceID Message

When this message will be sent:

This message MUST be sent by both PaC and PAA over an TLS connection right after a TLS handshake is finished.

Meaning of this message:

When a PaC or a PAA receives this message, it checks whether the Device Identifier contained in the message is the same as that is included in the MAC and/or IP header(s) encapsulating this message. If those two Device Identifiers are different, the receiver MUST return an Error message with Subtype "InvalidDeviceID" and immediately terminate the TLS session and the transport connection.

Structure of this message:

```
+----------+----------+-----------------+
| Type     | Subtype  | Length          |
|----------+----------+-----------------|
+----------+----------+-----------------|
|          |          | Device Identifier|
|----------+----------+-----------------|
```

Type

<table>
<thead>
<tr>
<th>Type value</th>
<th>Type name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x03</td>
<td>DeviceID</td>
</tr>
</tbody>
</table>

Subtype
Subtype value  Subtype name

0x01     IPAddress
0x02     IPAndMACAddresses

**IPAddress**

This Subtype is used when the sender has no MAC address associated with the transport connection.

**IPAndMACAddresses**

This Subtype is used when the sender has a MAC address associated with the transport connection.

**Length**

Variable (8, 16, 20 or 28).

**Payload**

When the Subtype value is 0x01 (IPAddress), either IPv4 or IPv6 address is included depending on whether the transport connection is carried over IPv4 or IPv6, respectively. When the Subtype value is 0x02 (IPAndMACAddresses), either IPv4 or IPv6 address is included depending on whether the transport connection is carried over IPv4 or IPv6, respectively, immediately followed by an IEEE EUI-64 address [EUI64].

### 4.11.4. AuthInfo Message

When this message will be sent:

This message MUST be sent during Non-TLS Full Authentication. This message MAY be sent during TLS Full Authentication. When this message is sent, it MUST be sent right after DeviceID message exchange. Multiple rounds of AuthInfo message exchange can occur until Success or Failure message is sent by the PAA.

**Meaning of this message:**

The contents and processing rules of the payload depends on the type of the authentication protocol.
Structure of this message:

<table>
<thead>
<tr>
<th>Type</th>
<th>Subtype</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>~ Authentication Protocol Message ~</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Type

<table>
<thead>
<tr>
<th>Type value</th>
<th>Type name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x04</td>
<td>AuthInfo</td>
</tr>
</tbody>
</table>

Subtype

<table>
<thead>
<tr>
<th>Subtype value</th>
<th>Subtype name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01</td>
<td>EAP</td>
</tr>
</tbody>
</table>

Length

Variable.

Payload

The contents depends on Subtype. If Subtype is 0x01 (EAP), an EAP PDU is included.

4.11.5. Success Message

When this message will be sent:

This message is sent by a PAA when a PaC is finally authenticated.

Meaning of this message:

This message means that the PaC that receives this message is finally authenticated and the device associated with the Device Identifier of the PaC is authorized for network access.

Structure of this message:
4.11.6. Failure Message

When this message will be sent:

This message is sent by a PAA when it finally fails to authenticate a PaC.

Meaning of this message:

This message means that authentication for the PaC that receives this message finally is not successful. When this message is sent,
the PAA MUST immediately terminate the TLS connection and the transport connection.

Structure of this message:

```
+---------------------------------+---------------------------------+-------------+
|     Type      |    Subtype    |             Length            |
+---------------------------------+---------------------------------+-------------+
```

Type

<table>
<thead>
<tr>
<th>Type value</th>
<th>Type name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x06</td>
<td>Failure</td>
</tr>
</tbody>
</table>

Subtype

<table>
<thead>
<tr>
<th>Subtype value</th>
<th>Subtype name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>NoSubtype</td>
</tr>
</tbody>
</table>

Length

The Length value is 4.

Payload

The payload is null.

4.11.7. Error Message

When this message will be sent:

This message is sent by a PAA or a PaC when it detects an error except for authentication failure. This message can occur at any time during PANA message exchange over a TLS connection.

Meaning of this message:

This message means that the sender detected an error except for authentication failure. The Subtype indicates the reason of the error. When this message is sent, the PAA MUST immediately terminate the TLS connection and the transport connection.

When the receiver of this message finds an error for this message, it MUST NOT return an Error message.
Structure of this message:

```
+-------------------+-------------------+-------------------+
<table>
<thead>
<tr>
<th>Type</th>
<th>Subtype</th>
<th>Length</th>
</tr>
</thead>
</table>
```

Type

<table>
<thead>
<tr>
<th>Type value</th>
<th>Type name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x07</td>
<td>Error</td>
</tr>
</tbody>
</table>

Subtype

<table>
<thead>
<tr>
<th>Subtype value</th>
<th>Subtype name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01</td>
<td>InvalidDeviceID</td>
</tr>
<tr>
<td>0x02</td>
<td>UnexpectedMessageType</td>
</tr>
<tr>
<td>0x03</td>
<td>UnsupportedMessageType</td>
</tr>
<tr>
<td>0x04</td>
<td>UnsupportedMessageSubtype</td>
</tr>
<tr>
<td>0x05</td>
<td>InvalidMessageLength</td>
</tr>
<tr>
<td>0x06</td>
<td>InvalidPayloadContents</td>
</tr>
<tr>
<td>0x07</td>
<td>HeartbeatResponseTimeout</td>
</tr>
</tbody>
</table>

InvalidDeviceID

This Subtype is used when an invalid Device Identifier is detected in a DeviceID message sent from the peer.

UnexpectedMessageType

This Subtype is used when a message is received with a Type value that is supported by the node but the type is different from that is expected at this specific protocol phase.

UnsupportedMessageType

This Subtype is used when a message is received with a Type value that is not supported.

UnsupportedMessageSubType

This Subtype is used when a message is received with an appropriate Type value but with a Subtype value that is not supported.

InvalidMessageLength

This Subtype is used when inconsistency is detected between the value of the Length field and the actual message length.

InvalidPayloadContents
This Subtype is used when the payload cannot be decoded based on the expected format defined for the specified message Type and Subtype.

HeartbeatResponseTimeout

This Subtype is used when the Heartbeat/Response timer for the PaC expires.

Length

The Length value is 4.

Payload

The payload is null.

4.11.8. Heartbeat Message

When this message will be sent:

This message will be exchanged between a PaC to a PAA after a Success message is sent by the PAA with a Subtype value 0x03 (DoAHP). The PAA sends a Heartbeat/Request message whenever it wants to check whether the PaC is active or not, and the PaC is requested to send Heartbeat/Response message back to the PAA in response to the Heartbeat/Request. Heartbeat/Request messages MAY be sent periodically.

Meaning of this message:

If the PAA that sent a Heartbeat/Request message to a PaC does not receive a Heartbeat/Response from the PaC for a Heartbeat/Response timeout period, the PAA MUST return an Error message with the Subtype "HeartbeatResponseTimeout" and immediately terminate the TLS connection and the transport connection.

Structure of this message:

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Subtype    |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Type

Type value  Type name
0x08        Heartbeat

Subtype

Subtype value  Subtype name
0x01          Request
0x02 Response
Length
The Length value is 4.
Payload
The payload is null.

5. Protocol Parameters

PortNumber
The destination port number to be used for the messages defined in this document. The same PortNumber is used for UDP, TCP and SCTP. The value of the PortNumber is TBD.

DefaultHeartbeatTimeout
The default timeout value for Heartbeat/Response message. The value of DefaultHeartbeatTimeout is 5 seconds.

All-PAA-Nodes
A link-local multicast address used for sending PAADiscover messages. In the case of IPv4, All-PAA-Nodes is the same as "all-hosts" group (224.0.0.1). In the case of IPv6, All-PAA-Nodes is a link-local scoped multicast address to be assigned by IANA.

6. Security Consideration

Potential security threats for PANA over TLS are discussed in this section.

6.1 Security on PAA Discovery

Since PAADiscover and AuthRequest messages are not authenticated, it is possible for an attacker to send those messages with bogus information.

The PAA that receives a bogus PAADiscover message will respond with an AuthRequest message. Since sending an AuthRequest message does not involve in any cryptographic computation or create any state at PAA, there is little impact on PAA for sending AuthRequest messages.

If the PAADiscover message has a bogus source address, then a PaC that is not the originator of the PAA Discover message may receive an
AuthRequest message, which may trigger Full Authentication or Fast Authentication. To reduce the impact of such a false authentication trigger, a PAA MAY have a policy for not accepting a new transport connection from a PaC device that has been authorized for network access until re-authentication becomes necessary for the PaC or that attempts to establish transport connections at a rate higher than the threshold value.

6.2 Security on Transport Connection for TLS

When TCP is used for the TLS transport, it is vulnerable to a blind masquerade attack (i.e., TCP SYN attack), which could let PAs spend memory resources for creating states for TCP connections. SCTP does not have such vulnerability due to the cookie-based four-way handshake mechanism.

Since transport protocol headers that envelop TLS PDUs are not protected, the headers are vulnerable to deliberate integrity attacks, which may incur data corruption for the transport protocol payload (blind attack is not possible). This kind of attacks are always detected by TLS anyway.

6.3 Security on TLS Handshake

The same security consideration as described in Appendix F of [TLS] is applied to this part. Since this protocol mandates the use of a server certificate for Full Authentication, man-in-the-middle attacks against contents carried over TLS connections are protected by TLS.

However, TLS handshake itself does not protect man-in-the-middle attacks against transport connections by spoofing MAC/IP address, unless client and server certificates are used for TLS handshake and each certificate is associated with either the IP address used for the transport connection or a DNS entry that is mapped to the transport IP address via secure DNS. DeviceID message exchange is used for protection against MAC/IP address spoofing (see next section).

6.4 Security on PANA Message Exchange over TLS Connection

Man-in-the-middle attacks against transport connections by spoofing MAC/IP address is prevented by using protected DeviceId message exchange over TLS.

EAP or other authentication protocol exchange encapsulated in AuthInfo messages are cryptographically protected by TLS. All EAP
messages including EAP-Response/Identity, EAP-Success and EAP-Failure messages can be encrypted and/or integrity protected. If the contents of EAP messages processed at a PAA need to be protected from being read in cleartext by the PAA, an appropriate EAP mechanism that supports EAP payload protection (i.e., EAP-SRP, EAP-TLS, EAP-TTLS, etc.) SHOULD be used.

PANA Success/Failure and Heartbeat messages are also cryptographically protected by TLS.

However, deliberate integrity attacks are possible at transport layer for these messages carried over TLS, as described in section "Security on Transport Connection for TLS".

7. Possible Future Direction

This section describes a possible future direction considering the ongoing work on EAP.

There are several EAP methods that use TLS for securing the payload of EAP messages. When those EAP methods are used, it might be possible to carry some of those messages in cleartext without compromising security at all. However, this requires a work in the EAP WG on security analysis as well as appropriate state machine definitions to make sure that only securing EAP payload is enough.

Once it is proven that only securing EAP payload is sufficient or the EAP specification is enhanced to have a method to protect both EAP header and payload, the PANA over TLS protocol will define an optional method that allows carrying AuthInfo messages without protection while other messages are still protected in order to strike a better balance between the required level of security and processing overhead.

Note that whatever EAP-based protection mechanism is applied to EAP header and/or payload, AuthInfo messages that carry EAP messages should be at least integrity protected if the PAA acts as a pass-through EAP authenticator so that an attacker cannot propagate "integrity broken" EAP messages all the way to the authentication server. If those messages are integrity protected by the PANA over TLS protocol, the PAA can immediately reject them before injecting into the backend authentication infrastructure.

8. Acknowledgments

The authors would like to thank Paal Engelstad for discussing on Device Identifier protection.
9. References


10. Authors’ Information

Yoshihiro Ohba
Toshiba America Research, Inc.
P.O. Box 136
Convent Station, NJ 07961-0136
USA
Phone: +1 973 829 5174
Email: yohba@tari.toshiba.com
Shinichi Baba
Toshiba America Research, Inc.
P.O. Box 136
Convent Station, NJ 07961-0136
USA
Phone: +1 973 829 4759
Email: sbaba@tari.toshiba.com

Subir Das
MCC 1D210R, Telcordia Technologies
445 South Street, Morristown, NJ 07960
Phone: +1 973 829 4959
email: subir@research.telcordia.com

Expires March, 2003