Protocol for Carrying Authentication for Network Access (PANA)
(draft-ietf-pana-pana-00.txt)

Authors:
Dan Forsberg
Yoshihiro Ohba
Basavaraj Patil
Hannes Tschofenig
Alper Yegin
Contents

• Introduction
• PAA Discovery
• Carrying EAP AVPs
• Creating a PANA SA
• ....
Producing the First Draft

- Design Team was established to work on initial proposal
- Work in progress:
  - Further discussions will be carried on the PANA ML

- Scope of the solution is bounded by:
  - draft-ietf-pana-usage-scenarios-04.txt
  - draft-ietf-pana-threats-eval-02.txt
  - draft-ietf-pana-requirements-04.txt

- Design team discussion archive available at:
  - http://danforsberg.info/pipermail/pana-dt

- Objective:
  - Satisfy the above requirements and scenarios by a simple protocol design
  - Various optimizations and enhancements left out for future consideration
Introduction: **PANA Framework**

Note: Some protocol interactions are optional.
Introduction: **PANA Protocol**

Interaction of PANA with the other protocols needs to be analyzed.
What was learned from the Usage Scenarios?

- PANA can be used in
  1. Environments with physical layer security
  2. Environments with link layer security
  3. Environments where no lower security is available

- Scenario (3) is the most difficult one for PANA deployment and adding the most requirements

- It is difficult to support all scenarios with a single protocol. Hence some protocol steps have to be optional.

- Multiple Authentication and Key Exchange methods should be supported ⇒ EAP
Assumptions

• **Topology Knowledge**
  Device Identifier information can be installed at the correct devices

• **Device Identifier Installation**
  Security provided by DI installation is sufficient for some environments. Otherwise, DI is accompanied by cryptographic keys.

• **Disconnect Indication**
  Link layer disconnect indication cannot be assumed

• **Session Key Establishment**
  Session key needs to be available for PANA SA

Note: Some assumptions will be explained in more details in subsequent slides.
PAA Discovery (1/2)

• Why?
  – To discover the PAA's address dynamically.

• How?
  – 1a) (Link local) multicast UDP packet from PaC.
  – 1b) PaC sends data packets.
    • EP sends a `PANA_discover` message to PAA, which contains PaC’s unicast address.
  – PAA sends `PANA_start` to PaC.
PAA Discovery (2/2)

• **Threats?**
  
  – Man-in-the-Middle between PaC and PAA.
  – DoS against PAA, DoS against PaC.

• **Countermeasures?**
  
  – Difficult since message exchange between neighboring nodes.
    • hop limit check
  
  – Goal:
    • Prevent off-path attacks (Cookie, Sequence numbers)
    • Prevent memory allocation with single message (Cookie)
Initial Sequence Number and Cookie

- Initial Sequence Number (ISN) mechanism is used to prevent blind DoS and off-path attacks.

- Cookie mechanism is used to prevent non-blind DoS attack.
  - Cookie is sent from PAA in \texttt{PANA\_start} message, but does not create any state in PAA that would enable DoS attack.
  - Cookie is implementation specific

- Message Flow

<table>
<thead>
<tr>
<th>PaC</th>
<th>PAA</th>
<th>Message(tseq,rseq) [AVPs]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1)</td>
<td>--</td>
<td>\texttt{PANA_discover}(0,0)</td>
</tr>
<tr>
<td>2)</td>
<td>&lt;</td>
<td>\texttt{PANA_start}(x,0) [Cookie]</td>
</tr>
<tr>
<td>3)</td>
<td>--</td>
<td>\texttt{PANA_start}(y,x) [Cookie]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(continued to authentication phase)</td>
</tr>
</tbody>
</table>
Carrying EAP over PANA

- **Why?**
  - Authentication and authorization required for network access procedures

- **How?**
  - EAP is payload of PANA (carried in **EAP AVP**)

- **Threats?**
  - MITM (injecting, modifying etc.); DoS; Eavesdropping

- **Countermeasures?**
  - Use an appropriate EAP method depending on the security requirements
  - Difficult to prevent all attacks until PANA SA is established
Carrying EAP over PANA
Transport Protocol Properties

• EAP requires ordered message delivery
  – EAP provides its own reliability and does not require the transport to be reliable

• EAP recommends EAP methods to provide message fragmentation
  – EAP TLS and PEAP support fragmentation, for example

• EAP supports retransmission for EAP Requests
  – Retransmission timeout calculation based on RFC2988 takes congestion control into account
Carrying EAP over PANA
Approach chosen by PANA

• PANA does not provide fragmentation.
  – Use EAP method fragmentation for EAP messages
  – Use IP fragmentation for other messages

• PANA provides:
  – Ordered delivery of EAP messages on top of UDP
  – Protection of PANA PDU after PANA SA is established
Carrying EAP over PANA
Sequence number handling (1/3)

• Why sequence number?
  – To provide ordered delivery of messages
  – Robustness against blind DoS attack is needed

• Considered approaches:
  – Single sequence number with PANA-layer retransmission
  – Dual sequence number with orderly-delivery
  – Dual sequence number with reliable-delivery

• Selected approach: Dual sequence number with orderly-delivery
  – Reason:
    • The 1st approach assumes ‘lock step’ messaging for all messages (EAP Success/Failure is not lock-step safe)
    • The 3rd approach is not simpler than the 2nd one

• Appendix in the draft provides detailed explanation
Carrying EAP over PANA
Sequence number handling (2/3)

- Following sequence numbers are included in PANA header
  - Transmitted sequence number (tseq)
  - Received sequence number (rseq)
- tseq starts from initial sequence number and is incremented by 1 when
  sending a message (even it is retransmitted)
- rseq is copied from the tseq field of the last accepted message
- When a message is received, it is valid (in terms of sequence #) if
  - Its tseq > tseq of the last accepted message, AND
  - Its rseq falls in the range
    [tseq of the last ack’ed msg+1, tseq of the last transmitted msg]
Carrying EAP over PANA
Sequence number handling (3/3)

PaC  PAA  Message(tseq,rseq)[AVPs]
--------------------
(continued from discovery and initial handshake phase)

<-----  PANA_auth(x+1,y)[EAP{Request}]
----->  PANA_auth(y+1,x+1)[EAP{Response}]
  .
  ...

<-----  PANA_auth(x+2,y+1)[EAP{Request}]
----->  PANA_auth(y+2,x+2)[EAP{Response}]

<-----  PANA_success(x+3,y+2)  // F-flag set
        [EAP{Success}, Device-Id, Data-Protection, MAC]
----->  PANA_success_ack(y+3,x+3)
        [Device-Id, MAC]  // F-flag set
PANA SA Establishment

• **Why?**
  – Protect subsequently exchanged PANA messages
    • E.g.: re-auth, disconnect
  – Bootstrap L2 or L3 access control, when needed

• **How?**
  – Key derived from EAP method; No algorithm negotiation

• **Threats?**
  – MITM - weak EAP methods

• **Countermeasures?**
  – Mutual authentication within EAP method
  – Weak EAP methods ⇒ see next slides
PANA SA Establishment

PAA Discovery

EAP Authentication (PaC ⇔ AAA[L|H] Authentication)

AAA Session Key Transport

Protected PANA Messages

PANA relies on EAP methods to produce keying material for PANA SA.

IETF56
PANA SA Establishment

- EAP method must provide session key for PANA SA

- There is no secure tunnel established between the PaC and the PAA (e.g. via ISAKMP or TLS) outside EAP!
EAP Method Choice

• PANA can carry any EAP authentication method

• It is the responsibility of the user and the network operator to pick the right method, depending on the environment
  – key derivation
  – mutual authentication
  – DoS resiliency

• PANA does not enable weak methods in insecure environments (a non-goal!)
  – PANA does not create a secure channel for them
  – PANA can carry EAP-tunneling methods (PEAP, EAP-TTLS)
    • Risk: MitM, needs to be fixed (not in PANA WG!)
Device ID Choice

- PaC will configure an IP address before PANA if it can
  - Network policy: EP might detect PaC’s attempts and trigger PANA first

- DI is either a link-layer address, or IP address
  - IP address: when PaC can configure one prior to PANA and IPsec is used for access control.
  - Link-layer address: otherwise.
Filter Rule Installation

• PANA protocol helps identifying who should gain access

• PAA helps EP build filters based on PANA results

• When PAA and EP are separated, a protocol is needed
  – This is not “PANA protocol”
  – PANA WG will “identify” at least one such protocol
  – MIDCOM WG’s output might be useful
Device Identifier Exchange

• **How?**
  – Key derived from EAP method; No algorithm negotiation

• **Why?**
  – By installing this device identifier unauthorized nodes are not able to inject packets.

• **Threats?**
  – MITM (injecting, modifying, etc.); DoS
  – IP spoofing; DI reuse (e.g. after roaming)

• **Countermeasures?**
  – Exchange data origin authenticated, replay and integrity protected with PANA SA
  – IP Spoofing and DI => see next slides

IETF56
Triggering a data protection protocol

• Why?
  – Spoofing attacks on shared links cannot be prevented by device ID based packet filters. Cryptographic protection needed.

• How?
  – PAA can signal if L2 or L3 ciphering should be initiated after PANA.
  – EAP established session key is indirectly used as an input to enforce link or network layer protection.
  – PANA can help bootstrap link-layer/network-layer ciphering
Re-authentication (1/3)

• **Why?**
  – Lower-layer disconnect indication is not always available
  – Garbage collection and stop of accounting required
  – Prevent DI spoofing and resulting service theft after disconnect (e.g. due to roaming)

• **How?**
  – Soft-state principle
  – Two types of re-authentication supported by PANA
    • Re-authentication based on **EAP**
    • Re-authentication based on **PANA_reauth/PANA_reauth_ack** exchange
  – Both PaC and PAA can initiate re-authentication
Re-authentication (2/3)

• **Threats?**
  – Spoofing re-authentication messages

• **Countermeasures?**
  – Protection by PANA SA
  – Limit re-authentication rate in implementation
Re-authentication (3/3)
Message Flow

<table>
<thead>
<tr>
<th>PaC</th>
<th>PAA</th>
<th>Message(tseq,rseq) [AVPs]</th>
</tr>
</thead>
<tbody>
<tr>
<td>----&gt;</td>
<td>PANA_reauth(q,p) [MAC]</td>
<td></td>
</tr>
<tr>
<td>&lt;-----</td>
<td>PANA_reauth_ack(p+1,q) [MAC]</td>
<td></td>
</tr>
</tbody>
</table>

Example Sequence for PaC-initiated Quick Re-authentication

<table>
<thead>
<tr>
<th>PaC</th>
<th>PAA</th>
<th>Message(tseq,rseq) [AVPs]</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;-----</td>
<td>PANA_reauth(p,q) [MAC]</td>
<td></td>
</tr>
<tr>
<td>----&gt;</td>
<td>PANA_reauth_ack(q+1,p) [MAC]</td>
<td></td>
</tr>
</tbody>
</table>

Example Sequence for PAA-initiated Quick Re-authentication
PANA session termination (1/2)

• **Why?**
  – PaC ⇒ PAA: Stop of accounting or finish network access
  – PAA ⇒ PaC: Many reasons (e.g. insufficient funds)

• **How?**
  – PANA message sent by PaC (disconnect indication)
  – PANA message sent by PAA (session revocation)
    • Revocation reason is included in **Revocation-Status AVP**

• **Threats?**
  – Adversary injecting a termination message (DoS)

• **Countermeasures?**
  – Protection by PANA SA
PANA session termination (2/2)
Message Flow

PaC  PAA  Message(tseq,rseq) [AVPs]

PANA_disconnect(q,p) [MAC]
PANA_disconnect_ack(p+1,q) [MAC]

Example Sequence for Disconnect Indication

PaC  PAA  Message(tseq,rseq) [AVPs]

PANA_revocation(p,q) [Revocation-Status,MAC]
PANA_revocation_ack(q+1,p) [MAC]

Example Sequence for Session Revocation
Open Issues and Next Steps

- **Discuss some open issues**
  - Flexibility of Device Identifier
  - Updating a device identifier
  - Session Identifier
  - Key derivation
  - Sequencing EAP methods
  - Integrity protection offered by PANA SA sufficient?
  - Re-authentication lifetime negotiation
  - Flow/Congestion Control

- **Next steps**
  - Improve draft
  - Define message formats
Backup Slides
Sequencing of EAP methods

• Why?
  – Some scenarios require more sequencing of EAP methods

• How?
  – Multiple EAPs performed in a single PANA session
    • Each EAP is delimited with PANA_success/failure
    • PANA_success/failure has F-flag to indicate final exchange.
Sequencing of EAP methods
Message Flow

(continued from discovery and initial handshake phase)

// First EAP run for NAP authentication

<----- PANA_auth[EAP{Request}]

-----> PANA_auth[EAP{Response}]

<----- PANA_success[EAP{Success},MAC]  // F-flag not set

-----> PANA_success_ack[Device-ID, MAC]  // F-flag not set

// Second EAP run for ISP authentication

<----- PANA_auth[EAP{Request},MAC]

-----> PANA_auth[EAP{Response},MAC]

<----- PANA_success[EAP{Success}, MAC]  // F-flag set

-----> PANA_success_ack[MAC]  // F-flag set
Session key for PANA SA is a combination of two AKA steps.