Revised Cookie Processing in IKEv2

draft-smyslov-ipsecme-ikev2-cookie-revised

Valery Smyslov
svan@elvis.ru

IETF 115
Using Cookies in IKEv2

According to RFC 7296, the most recent `IKE_SA_INIT` request is included in the `AUTH` payload calculation in the `IKE_AUTH` exchange. In this example it is `req2` for both the initiator and the responder.
Problem Scenario 1

**Initiator**

- req1  **IKE_SA_INIT**
  HDR,SAi1,KEi,Ni

- req1 (resend)  **IKE_SA_INIT**
  HDR,SAi1,KEi,Ni

- req2  **IKE_SA_INIT**
  HDR,N(COOKIE),SAi1,KEi,Ni

- req3  **IKE_AUTH**
  HDR,SK{IDi,[CERT,][CERTREQ,][IDr,] AUTH, SAi2, TSi, TSr}

**Responder**

- Under attack
  - resp1  **IKE_SA_INIT**
    HDR,N(COOKIE)

- No more under attack
  - resp2  **IKE_SA_INIT**
    HDR,SAr1,KEr,Nr,[CERTREQ,]

- X

- resp3  **IKE_AUTH**
  HDR,SK{N(AUTHENTICATION_FAILED)}

The most recent **IKE_SA_INIT** request sent by the initiator is **req2**, while the responder only received **req1**, so authentication would fail.
Problem Scenario 2

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td>req1 <strong>IKE_SA_INIT</strong> HDR,SAi1,KEi,Ni</td>
<td>resp1 <strong>IKE_SA_INIT</strong> HDR,N(COOKIE, \textit{c1})</td>
</tr>
<tr>
<td>req1 (resend) <strong>IKE_SA_INIT</strong> HDR,SAi1,KEi,Ni</td>
<td>Under attack, cookie secret changed</td>
</tr>
<tr>
<td>req2 <strong>IKE_SA_INIT</strong> HDR,N(COOKIE, \textit{c2}),SAi1,KEi,Ni</td>
<td>resp2 <strong>IKE_SA_INIT</strong> HDR,N(COOKIE, \textit{c2})</td>
</tr>
<tr>
<td>req3 <strong>IKE_SA_INIT</strong> HDR,N(COOKIE, \textit{c1}),SAi1,KEi,Ni</td>
<td><strong>X</strong></td>
</tr>
<tr>
<td>req4 <strong>IKE_AUTH</strong> HDR,SK{IDi,[CERT,][CERTREQ,]} [IDr,] AUTH, SAi2, TSi, TSr}</td>
<td>resp4 <strong>IKE_AUTH</strong> HDR,SK{N(AUTHENTICATION FAILED)}</td>
</tr>
</tbody>
</table>

The most recent **IKE_SA_INIT** request sent by the initiator is \textit{req3}, while the responder only received \textit{req2}, so authentication would fail.
Source of the Problem

- The `IKE_SA_INIT` request can be sent several times with different content depending on the responder state.
- If there is a high probability of packets loss and reordering, then peers may complete the `IKE_SA_INIT` exchange having different views on what was the most recently sent `IKE_SA_INIT` request.
- This request message is used in calculation of the `AUTH` payload. If peers use different messages for the calculation, the authentication would erroneously fail.
Severity of the Problem

• There are some preconditions for this problem to become noticeable
  – network with high probability of packet loss and delay
  – relatively frequent change of responder state (either changing cookie generation secret or changing responder’s mind whether it is under attack)
• It might be extremely rare in normal conditions, but in stress tests we observed that up to 5% of SAs failed due to this problem
  – customers wonder why authentication sometimes fails with proper credentials
• This is a protocol flaw
Proposed Solution Overview

- Revise cookie processing by excluding Notify payload containing cookie (if present) from the \texttt{IKE\_SA\_INIT} request message when calculating the \texttt{AUTH} payload content
  - the cookie is already verified by the responder, no need to include it into the data to be authenticated
- For backward compatibility make the revised processing negotiable
Negotiation

Responder includes a new notification \texttt{REVISED_COOKIE} in the message containing \texttt{COOKIE} notification. If initiator also supports this extension, it returns cookie in this notification instead of \texttt{COOKIE} notification.
Revised Cookie Processing

• If peers agreed upon using this extension then the cookie processing is changed
  – no changes in cookie anti-clogging function – responder still sends stateless cookie and when it is returned back by initiator it MUST be verified before message is processed

According to RFC7296 initiator’s `AUTH` payload is calculated by signing (or MAC’ing) the blob:

`InitiatorSignedOctets = RealMessage1 | NonceRData | MACedIDForI`

– if `COOKIE Notify` payload is present in `RealMessage1` (i.e. in `IKE_SA_INIT` request message), then for the purpose of `AUTH` payload calculation the message is modified as if it contained no this payload
Adjusting IKE_SA_INIT Request for AUTH Payload Calculation

<table>
<thead>
<tr>
<th>IKE SA Initiator's SPI</th>
<th>IKE SA Responder's SPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>NextPld1</td>
<td>Version</td>
</tr>
<tr>
<td>Message ID</td>
<td></td>
</tr>
<tr>
<td>MsgLen</td>
<td></td>
</tr>
<tr>
<td>NextPld2</td>
<td>RESERVED</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Cookie</td>
<td></td>
</tr>
<tr>
<td>Rest of Message</td>
<td></td>
</tr>
</tbody>
</table>

IKE SA Initiator's SPI
IKE SA Responder's SPI
NextPld2
Version
Exchange
Flags

讯息 ID

MsgLen’ = MsgLen - PldLen1

Rest of Message

COOKIE Notify Payload
Thanks

• Comments? Questions?
• Is this problem worth to address?
• Is the suggested approach reasonable?
• WG adoption?