Motivation

Humans are using the Internet
... are authenticated
... are access controlled
Humans

Authenticated by
... by passwords
... by OTPs
... by Token cards
NOTE!

We are only piggybacking existing authentication protocol
HIP BEX

Initiator

I1: trigger exchange
------------------------>
select precomputed R1

R1: puzzle, D-H, key, sig
<-------------------------
check sig
remain stateless
solve puzzle

I2: solution, D-H, {key}, sig
------------------------>
compute D-H
check puzzle
check sig

compute D-H
check sig
check sig

Responder
EAP

- Lock-step protocol
- No fragmentation support
- Initiated by the authenticator
- Both peers may be authenticators simultaneously
## EAP negotiation

<table>
<thead>
<tr>
<th>Supplicant</th>
<th>Authenticator</th>
<th>Authentication Server</th>
</tr>
</thead>
</table>
| -- Start --------------------------> | -- Response identity -------> | <- Request 1 -------------->
| <- Request identity --------------- | <- Request 1 --------------- | -- Response 1 -----------------> |
| -- Response identity -------------- | -- Response 1 ----------------> | . |
| <- Request 1 ---------------------- | -- Request n --------------- | . |
| -- Response 1 --------------------- | -- Request n --------------- | . |
| . | . | . |
| <- Request n ---------------------- | <- Request n --------------- | . |
| -- Response n --------------------- | -- Response n --------------- | . |
| <- Success ------------------------ | <- Success ------------------ | . |
Road to here...

- Extending BEX:
  - Reuse control packets → dirty / state machine
  - Hiccups in front → logical / state machine
  - Hiccups in the middle → dirty / state machine
  - Hiccups after bex → just wrong

- Access control the SA and use UPDATEs
EAP_SIGNED / EAP_UNSIGNED parameters

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAP (Variable length)</td>
<td>/</td>
</tr>
</tbody>
</table>

Type          SIGNED TBD-IANA (998) and UNSIGNED TBD-IANA (632997)
Length        Length in octets, excluding Type, Length, and Padding
EAP           EAP message, variable length.
Padding       Any Padding, if necessary, to make the parameter a multiple of 8 bytes as defined in [RFC5201] in section 5.2.1.
EAP Challenge/Response

Initiator               Responder

I1                      Select precomputed R1
-------------------------------
R1: puzzle,
    key, sig, SERVICE_OFFER,
    EAP_UNSIGNED(challenge)
<-------------------------------
Check sig                Remain stateless
Solve puzzle
Calculate response

I2: solution,
    EAP_SIGNED(response),
    SERVICE_ACK, {key}, sig
-------------------------------
Compute D-H              Check solution
Check solution
Check sig
Check response

R2: EAP_SIGNED(success), sig
<-------------------------------
Check sig                Calculate session key
Calculate session key
Change traffic
Change traffic restrictions

ESP
<-------------------------------

HIIT, Samu Varjonen 10
Middlebox in challenge/response

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Middlebox</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1,</td>
<td>Add EAP_UNSIGNED(*) and SERVICE_OFFER</td>
<td>I1,</td>
</tr>
<tr>
<td></td>
<td>Verify response</td>
<td>R1</td>
</tr>
<tr>
<td>R1, EAP_UNSIGNED(*), SERVICE_OFFER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I2, EAP_SIGNED(**), SERVICE_ACK</td>
<td>Change traffic restrictions</td>
<td>I2, EAP_SIGNED(**), SERVICE_ACK</td>
</tr>
<tr>
<td></td>
<td>Add response</td>
<td>R2</td>
</tr>
<tr>
<td>R2,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* = Challenge
** = Response
EAP-pwd Handshake

EAP peer

EAP-pwd-ID/Request

EAP-pwd-ID/Response

EAP-pwd-Commit/Request

EAP-pwd-Commit/Response

EAP-pwd-Confirm/Request

EAP-pwd-Confirm/Response

EAP Success

EAP server
HIP-EAP-pwd

Initiator
(EAP peer)

Responder
(EAP server)

I1
------------------------>
Select precomputed R1

R1: puzzle, SERVICE_OFFER,
key, sig

Check sig
Solve puzzles

I2: solution, SERVICE_ACK,
{key}, sig

Compute D-H
------------------------>
Check solution
Check sig

R2: sig,
EAP_SIGNED(pwd-ID/Request)

Check sig

UPDATE:
EAP_SIGNED(pwd-ID/Response)

UPDATE:
EAP_SIGNED(pwd-Commit/Request)

UPDATE:
EAP_SIGNED(pwd-Commit/Response)

UPDATE:
EAP_SIGNED(pwd-Confirm/Request)

UPDATE:
EAP_SIGNED(pwd-Confirm/Response)

UPDATE: EAP_SIGNED(Success)

Change traffic restrictions

ESP

------------------------>
Middlebox in HIP-EAP-pwd

Initiator | Middlebox | Responder
---|---|---
I1, | | I1,
R1, SERVICE_OFFER | Add SERVICE_OFFER | R1
I2, SERVICE_ACK | Verify response | Let I2 through | I2, SERVICE_ACK
R2, EAP_US(pwd-ID/*) | Add EAP_US(pwd-ID/*) | R2
| | | UPD,
UPD, EAP_S(pwd-ID/**), SEQ | | | UPD,
| | | UPD, ACK, EAP_US(pwd-Commit/*)
| | | | | | UPD, ACK
| | | | | | UPD, SEQ
| | | | | | UPD, ACK
| | | | | | UPD, SEQ
| | | | | | Change traffic restrictions
| | | | | | Add EAP response
UPD, ACK, EAP_US(success) | Add EAP response | UPD, ACK

* = Challenge
** = Response
EAP_S = EAP_SIGNED
EAP_US = EAP_UNSIGNED

HIIT, Samu Varjonen
Thanks

Questions?