LINUX FULL IPSEC OFFLOAD
Linux IPsec Workshop, March 2018
## IPsec Offload Modes

<table>
<thead>
<tr>
<th>Crypto Offload</th>
<th>Full IPsec Offload (new)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current IPsec offload support in xfrm</td>
<td>Topic for Discussion</td>
</tr>
<tr>
<td><strong>SW</strong></td>
<td></td>
</tr>
<tr>
<td>IPsec encap/decap</td>
<td></td>
</tr>
<tr>
<td>Padding insertion/validation</td>
<td></td>
</tr>
<tr>
<td>Anti-replay</td>
<td></td>
</tr>
<tr>
<td>Counters updates</td>
<td></td>
</tr>
<tr>
<td>SPD</td>
<td></td>
</tr>
<tr>
<td>Encrypt/Decrypt/Integrity</td>
<td>IPsec encap/decap</td>
</tr>
<tr>
<td></td>
<td>Encrypt/Decrypt/Integrity</td>
</tr>
<tr>
<td></td>
<td>Padding insertion/validation</td>
</tr>
<tr>
<td></td>
<td>Anti-replay</td>
</tr>
<tr>
<td></td>
<td>Counter updates</td>
</tr>
<tr>
<td></td>
<td>SPD</td>
</tr>
</tbody>
</table>
IPsec Full Offload – Discussion Areas

• Handling differing device capabilities
• Exception Handling / SW Fall-back
• Padding
• SA Lifetimes
• Counters
• SPD offload
IPsec Full Offload – Device Capabilities

- How to distinguish between full offload and encrypt/decrypt offload modes?
- Use existing NETIF_F_HW_ESP & use xfrm_state_offload.flags to indicate different offload modes?
- Alternative: XFRM user specifies desired offload mode?
- Alternative: ???
- Device offload capability check as part of xdo_dev_state_add
IPsec Full Offload - Behaviour on Exceptions?

Software Fall-back
- SA miss
- Fragmentation?

Drop Packet
- ICV validation failure
- Anti-replay check failure
- Packet length errors
- Tx packet size exceeds MTU after encap
- Fragmentation
IP Reassembly – Anti-Replay

- Fragments sent to SW for Reassembly + IPsec
- Non-Fragmented packets processed in HW
- Reassembly latencies may cause reassembled packet to fall outside of the anti-replay window.

Max Reassembly Time (usec) Before ARW failures

<table>
<thead>
<tr>
<th>ARW Size -&gt;</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1024</th>
<th>4096</th>
</tr>
</thead>
<tbody>
<tr>
<td>10Gbps</td>
<td>10.9</td>
<td>21.7</td>
<td>43.4</td>
<td>86.8</td>
<td>347.3</td>
</tr>
<tr>
<td>40Gbps</td>
<td>2.7</td>
<td>5.4</td>
<td>10.9</td>
<td>21.7</td>
<td>86.8</td>
</tr>
<tr>
<td>100Gbps</td>
<td>1.1</td>
<td>2.2</td>
<td>4.3</td>
<td>8.7</td>
<td>34.7</td>
</tr>
</tbody>
</table>

Ideal: Reassemble in HW or Drop Fragments

![Diagram of IP Reassembly and Anti-Replay](image-url)
IPsec Full Offload – Padding, Lifetime & Counters

**IPsec Padding**
- Insert minimum required padding on Tx
- Validate and strip padding on Rx
- TFC?

**SA Lifetime Limits**
- Byte & Packet count, Elapsed time
- SW vs HW split? Configurable?
- HW generated events on reaching soft/hard limit

**Per SA Counters:**
- Integrity failures
- Replay failures
- Replay window failures
- Pkts processed?
- Bytes processed?

New xdo callback required to retrieve counters from device.
IPsec Full Offload – SPD Policy

Policy offloaded in all full offload use cases?
Policy selector complexity? Fully compatible with xfrm_selector
IPsec Full Offload – sk_buff

Reuse existing xfrm_offload status & flags to indicate processing performed