1.6.1 Stack sealing hotfix explanation
Conclusion

- The SEAL position is shifted 3 words upper than expected (Bottom).
- But this won’t bring damages or security issues, not even functionality errors.
- A coding mistake needs to be fixed – the fix has been merged.
Background: Why seal?

ldr r0, = ns_func
  -> blxn r0

ns_returned:
  cmp r0, #0

ns_func: ;LR = FNC_RETURN
  ...
  -> bx lr

ns_returned:
  ...
  bx lr

<- xSP_S

<- xSP_S
Background: Why seal and the answer

What will happen if NS just performs ‘BX LR(FNC_RETURN)’ when it is not in a S to NS calling procedure?

This content will be treated as the secure function address.

The solution is to apply SEALs at these two word’s place – the SEAL is not a valid function address hence causes exceptions after fetched.

Why no sealing in v1.6.0 is still safe

- The advisory requires to apply sealing on each stack for applicability, in fact not all scenarios is the same as the expectations in advisory.
- Stack ER_INITIAL_PSP is designed for interacting with NS, it launches NSPE by BLXNS hence there is always valid return address on the stack.
- This valid return address points to a panic, hence NSPE has no chance to tamper SPE execution by BX FNC_RETURN.

```
1dr r0, = ns_boot
-> blx r0
ns_returned:
  panic
```

<table>
<thead>
<tr>
<th>Expected</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>ns_returned</td>
<td>ns_returned</td>
</tr>
<tr>
<td>RET_PSR</td>
<td>RET_PSR</td>
</tr>
<tr>
<td>SEAL</td>
<td>SEAL</td>
</tr>
<tr>
<td>SEAL</td>
<td></td>
</tr>
</tbody>
</table>

<- xSP_S  <- xSP_S
Thank You
Danke
Gracias
Grazie
谢谢
ありがとう
ありがとう
감사합니다
धन्यवाद
شكرًا
ধন্যবাদ
תודה