

arm

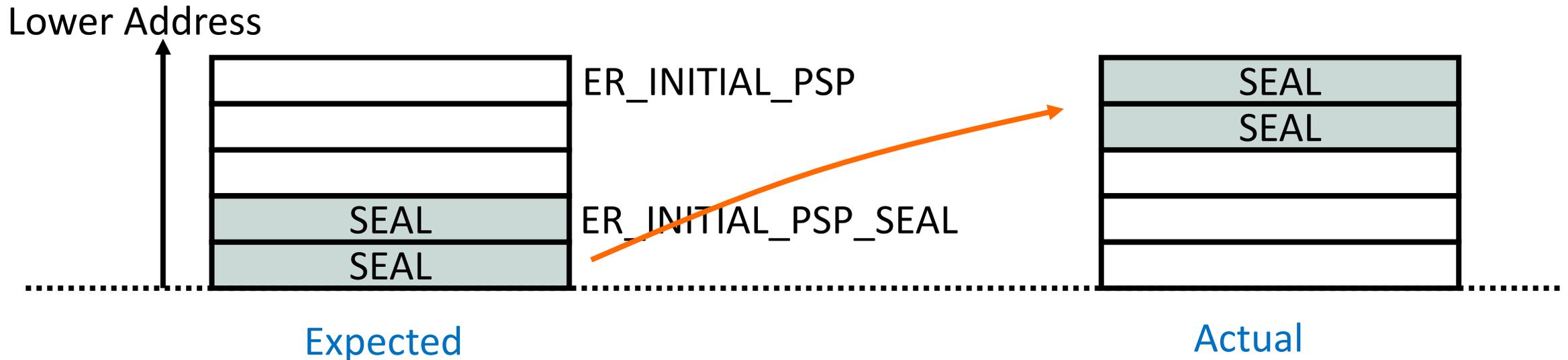
1.6.1 Stack sealing hotfix explanation

Ken Liu
Nov 30th

© 2022 Arm

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?

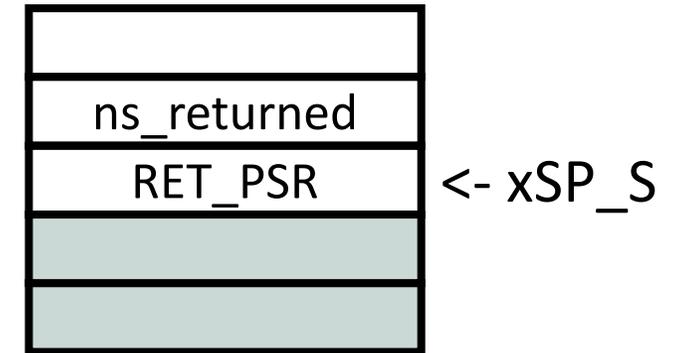
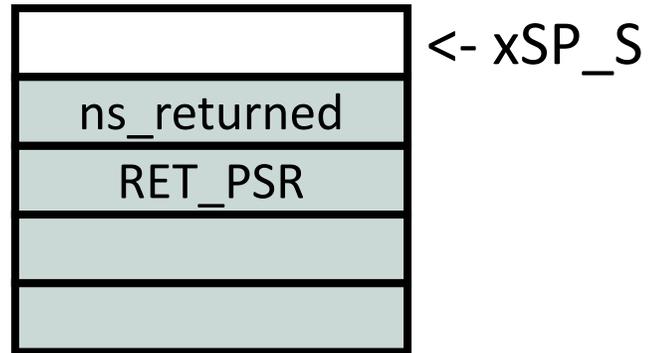
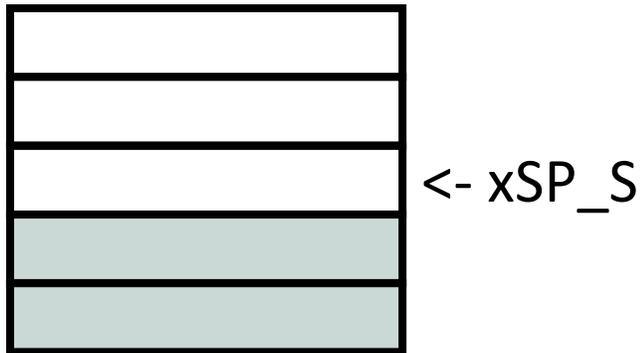
```
ldr r0, = ns_func  
-> blxns r0  
ns_return:  
  cmp r0, #0
```

```
ldr r0, = ns_func  
blxns r0  
ns_returned:  
  cmp r0, #0
```

```
ldr r0, = ns_func  
blxns r0  
ns_returned:  
-> cmp r0, #0
```

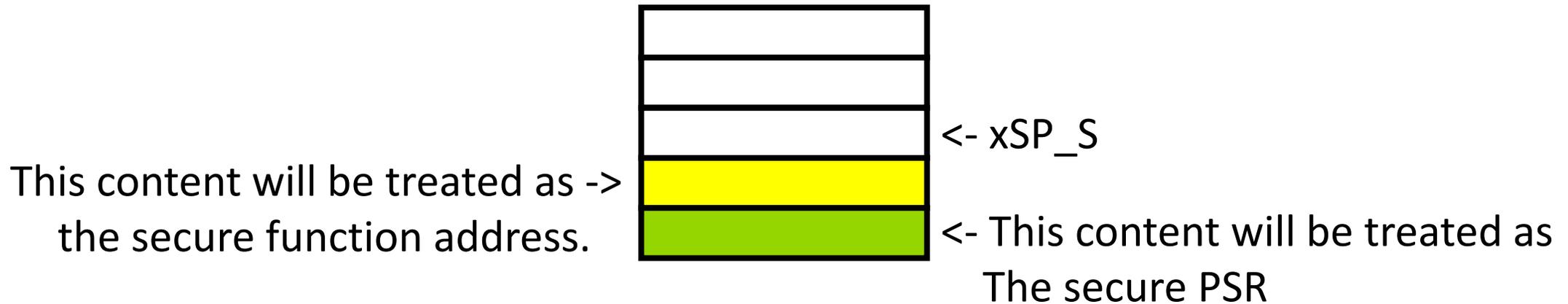
```
ns_func: ;LR = FNC_RETURN  
  ...  
-> bx lr
```

```
ns_func: ;LR = FNC_RETURN  
  ...  
  bx lr
```



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?



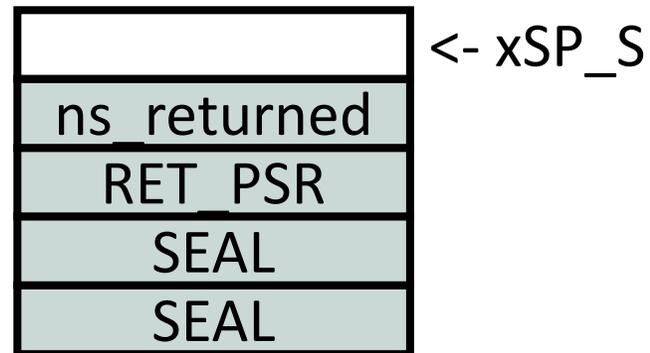
- + The solution is to apply SEALS at these two word's place – the SEAL is not a valid function address hence causes exceptions after fetched.

<https://developer.arm.com/Arm%20Security%20Center/Armv8-M%20Stack%20Sealing%20Vulnerability>

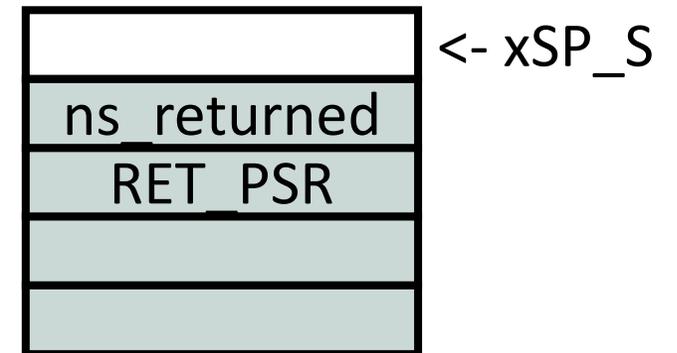
Why no sealing in v1.6.0 is still safe

- ✦ 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.

```
ldr r0, = ns_boot  
-> blx r0  
ns_returned:  
panic
```



Expected



Actual

arm

Thank You

Danke

Gracias

Grazie

谢谢

ありがとう

Asante

Merci

감사합니다

धन्यवाद

Kiitos

شكرًا

ধন্যবাদ

תודה