Réunion flash

Point hebdomadaire

Duzes Florian

Sommaire

01
État des lieux

Point actuel

Fait

- Méthodologie en x86_64
- 1ère méthodologie en ARM

En cours

- Méthodologie ARM
- Étendre les scripts pour le dossier */tests* en ARM

Prévus
Pour le 15/06

- Automatisation des .ini
- Couvrir les tests
- Couvrir les primitives Hacl*
- RiskV - compil à la main de tests

Réalisation

Fait

- Journée d'intégration Inria avec les RH
- Construction de la chaîne de compilation
- Balade pour Risk-V sur : https://riscv.org/

En cours

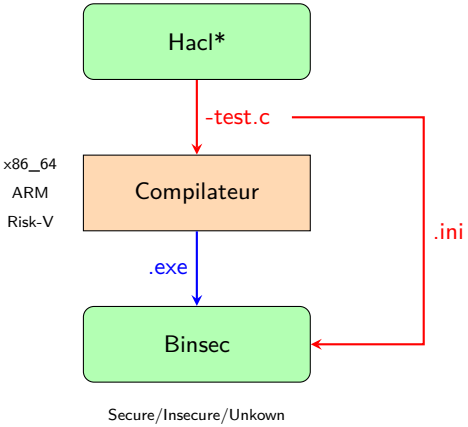
- Méthodologie ARM
- Étendre les scripts pour le dossier */tests* en ARM
- Automatisation de x86_64

Prévus
Pour le 15/06

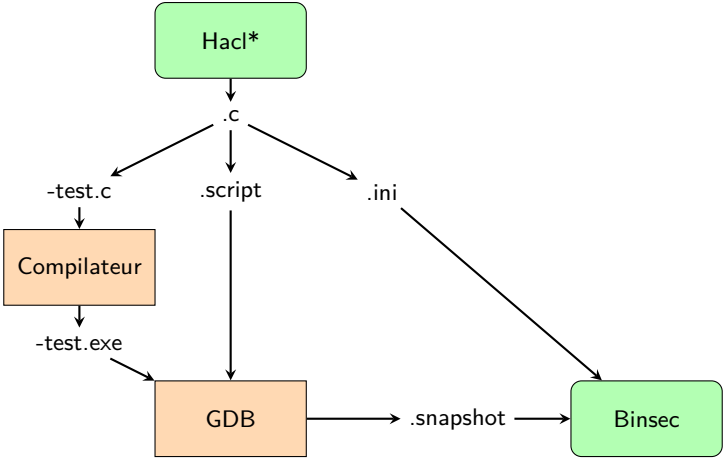
- Automatisation des .ini
- Couvrir les tests
- Couvrir les primitives Hacl*
- RiskV - compil à la main de tests

02
Automatisation

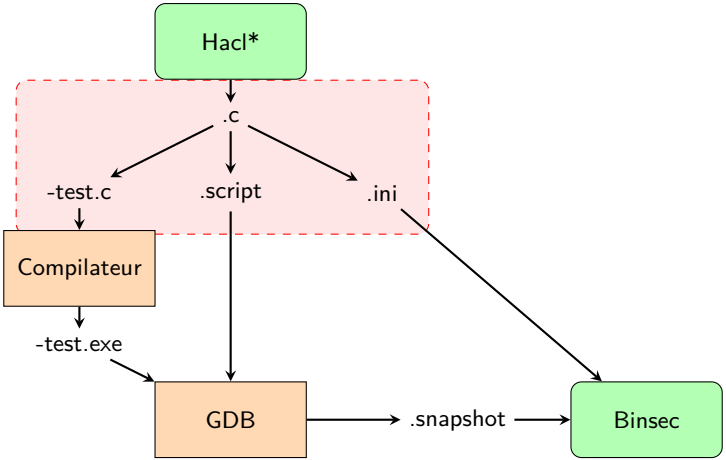
La chaîne de compilation



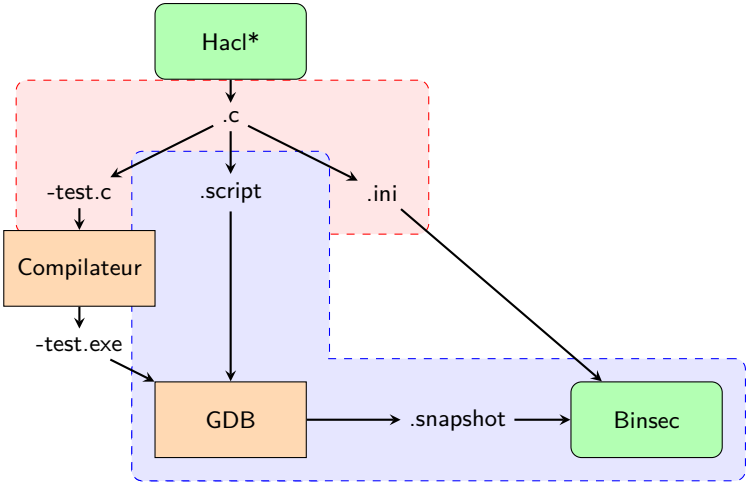
Spécificité de x86_64



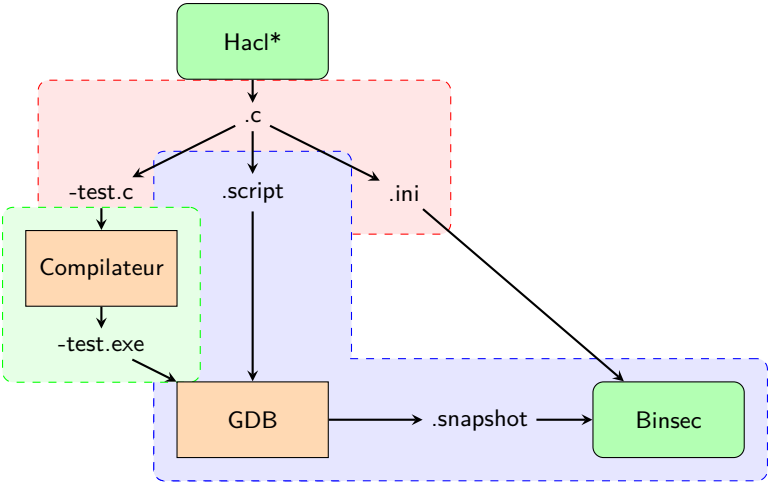
Spécificité de x86_64



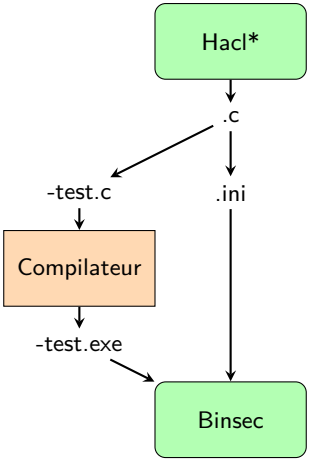
Spécificité de x86_64



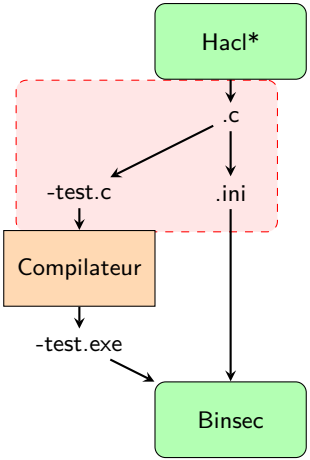
Spécificité de x86_64



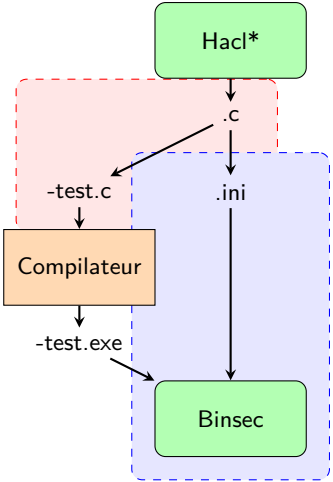
Spécificité de ARM



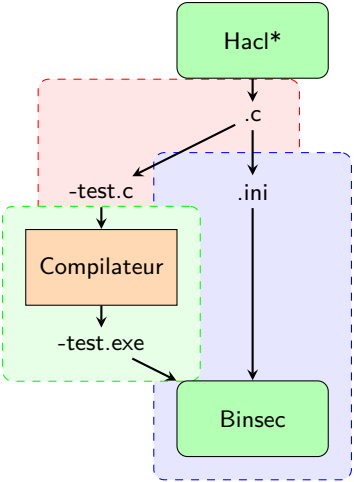
Spécificité de ARM



Spécificité de ARM



Spécificité de ARM



03
Protocole ARM

Point de départ

```
#fin de la zone de .text
#@[sp, 8] := 0x00404860 as return_address

#fin arbitraire : fin de calculs
@[sp, 8] := 0x004005c0 as return_address

# load common sections from ELF file
load sections .plt, .text, .rodata, .data, .got, .got.plt, .bss from file

secret global plain, aead_aad, aead_key, aead_nonce

starting from <main>
with concrete stack pointer

halt at return_address
explore all
```

Code 1 – script.ini

Problème et solution

Insatisfaisant

- Arrêt prompt de l'analyse
- Couverture totale probable

Problème et solution

Insatisfaisant

- Arrêt prompt de l'analyse
- Couverture totale probable

Solution de l'adressage

```
%.exe: %.o
  $(CC) $(CFLAGS) $(LDFLAGS) $^ ../dist/gcc-compatible/
      libevercrypt.a -static -o $@
```

Code 3 – tests/Makefile

Adaptation

```
Section de réadressage '.rela.plt' à l'adresse de décalage 0x1b0 contient 7 entrées :
Décalage         Info             Type             Val.-symboles Noms-symb.+ Addenda
00000048d000  000000000408 R_AARCH64_IRELATI                    419060
00000048d008  000000000408 R_AARCH64_IRELATI                    4193e0
00000048d010  000000000408 R_AARCH64_IRELATI                    438020
00000048d018  000000000408 R_AARCH64_IRELATI                    419170
00000048d020  000000000408 R_AARCH64_IRELATI                    418570
00000048d028  000000000408 R_AARCH64_IRELATI                    438020
00000048d030  000000000408 R_AARCH64_IRELATI                    418570
```

Code 4 – readelf -r

Solution ?

```
load sections .plt, .text, .rodata, .data, .got, .got.plt, .bss from file

secret global plain, aead_aad, aead_key, aead_nonce

@[0x00000048d000 ,8] := <__memmove_generic>
@[0x00000048d008 ,8] := <__memcpy_generic>
@[0x00000048d010 ,8] := <__memchr_generic>
@[0x00000048d018 ,8] := _dl_aarch64_cpu_features
@[0x00000048d020 ,8] := <__memcpy_thunderx2>
@[0x0000004002ac ,8] := <__memchr_generic>
@[0x00000048d024 ,8] := <__memchr_generic>
@[0x00000048d028 ,8] := <__memchr_generic>
@[0x00000048d030 ,8] := <__memcpy_thunderx2>

lr<64> := 0xdeadbeef as return_address

starting from <main>
with concrete stack pointer

halt at return_address
explore all
```

Code 5 – script.ini

Presque

```
[sse:debug] 0x00402b58 stp      q13, q11, [sp,#336]     # <Hacl_Chacha20_Vec128_chacha20_encrypt_128> + 0x1628
[sse:debug] 0x00402b5c str      q9, [sp, #368]          # <Hacl_Chacha20_Vec128_chacha20_encrypt_128> + 0x162c
[sse:debug] 0x00402b60 bl       0x4002a0                # <Hacl_Chacha20_Vec128_chacha20_encrypt_128> + 0x1630
[sse:debug] 0x004002a0 adrp     x16, 0x48d000
[sse:debug] 0x004002a4 ldr      x17, [x16,#24]
[sse:debug] 0x004002a8 add      x16, x16, #0x18
[sse:debug] 0x004002ac br       x17
[sse:info] Empty path worklist: halting ...
[sse:warning] Enumeration of jump targets @ 0x004002ac hit the limit 3 and may be incomplete
[sse:warning] Cut path 3 (non executable) @ 0x1c3a9965
[sse:warning] Cut path 2 (non executable) @ 0xffffffffffffffff
[sse:warning] Cut path 1 (non executable) @ 0xfffffffffffffffe
[sse:warning] Threat to completeness :
            - some jump targets may have been omitted (-sse-jump-enum)
[checkct:warning] Exploration is incomplete:
                - 3 paths fell into non executable code segments
                - some jump targets may have been omitted (-sse-jump-enum)
```

Code 6 – binsec -sse -sse-depth 1000000 -sse-script study.ini -checkct chacha20poly1305-128-binsec-test.exe

Si je rajoute des stops

```
[sse:warning] Enumeration of jump targets @ 0x004002ac hit the limit 3 and may be incomplete
[sse:warning] Cut path 2 (non executable) @ 0xffffffffffffffff
[sse:warning] Cut path 1 (non executable) @ 0x00000000
[sse:warning] Threat to completeness :
              - some jump targets may have been omitted (-sse-jump-enum)
[checkct:warning] Exploration is incomplete:
                  - 2 paths fell into non executable code segments
                  - some jump targets may have been omitted (-sse-jump-enum)
                  - 9 SMT solver queries remain unsolved (-fml-solver-timeout)
```

Code 7 – binsec -sse -sse-depth 1000000 -sse-script study.ini -checkct chacha20poly1305-128-binsec-test.exe

04
Conclusion

Conclusion

Corrections pour ARM

- Fix les erreurs de compilations
- Poursuivre la conception des scripts à la main sur les tests

Conclusion

Corrections pour ARM

- Fix les erreurs de compilations
- Poursuivre la conception des scripts à la main sur les tests

Automatisation de x86_64

- Partir des tests présents ?
- Compiler depuis le code source C/F* ?

Merci.