Les aléas d'un ransomware

Yoann Guillot

ANSSI

2020-06-04

Plan

L'attaque

- Un cryptolocker de plus
- L'attaquant s'installe dans le SI
- Ciblage de l'environnement
  - Sauvegardes
  - Applications métier
- Chiffrement des machines

Plan

Chiffrement d'un fichier

- Génération d'une clé RC4 par fichier
  - Aléatoire
  - 117 octets
- Chiffrement du fichier

Hélas, l'aléa s'est laissé aller

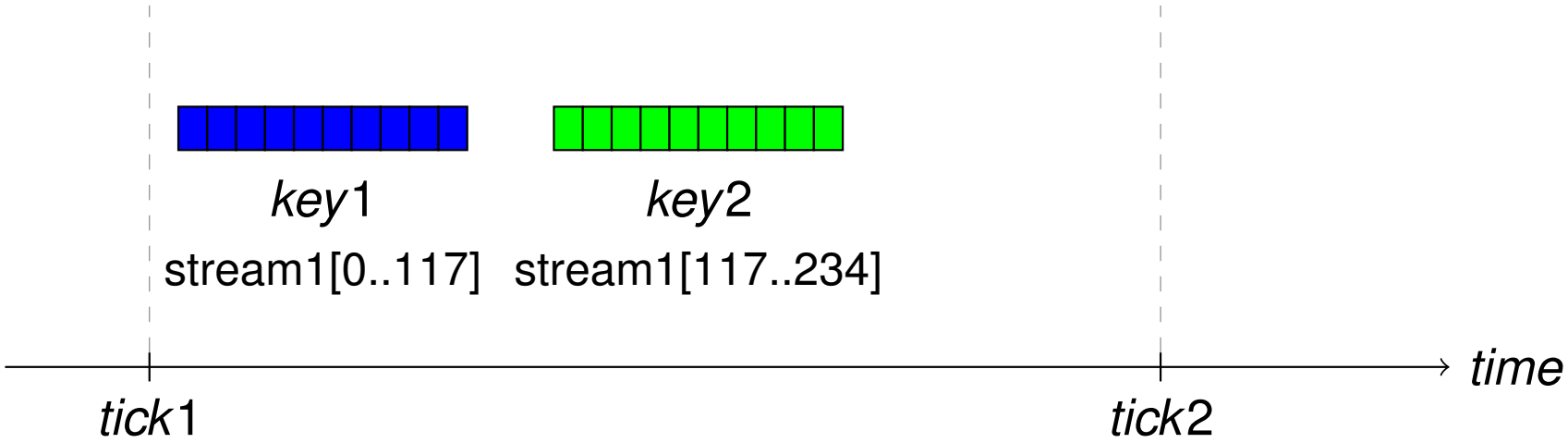
GetRandByte() :

- tick = GetTickCount()
- if tick != lasttick
  - PRNG_reseed(tick)
  - lasttick = tick
- return PRNG_nextbyte()

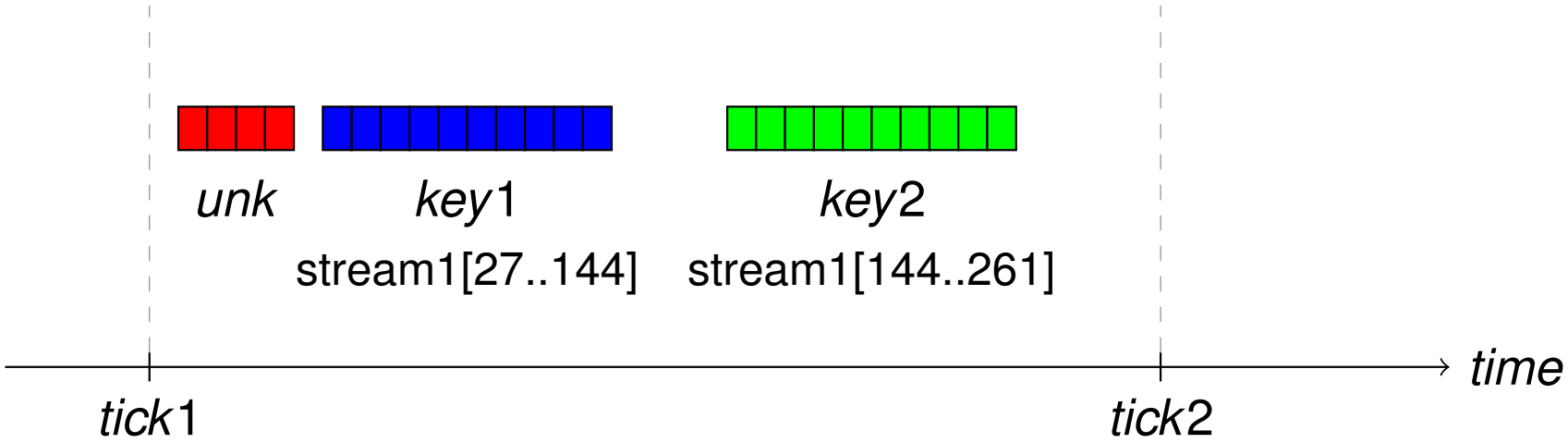
GetTickCount()

- API standard Windows
- Nombre de millisecondes depuis le boot
- Valeur sur 32-bit

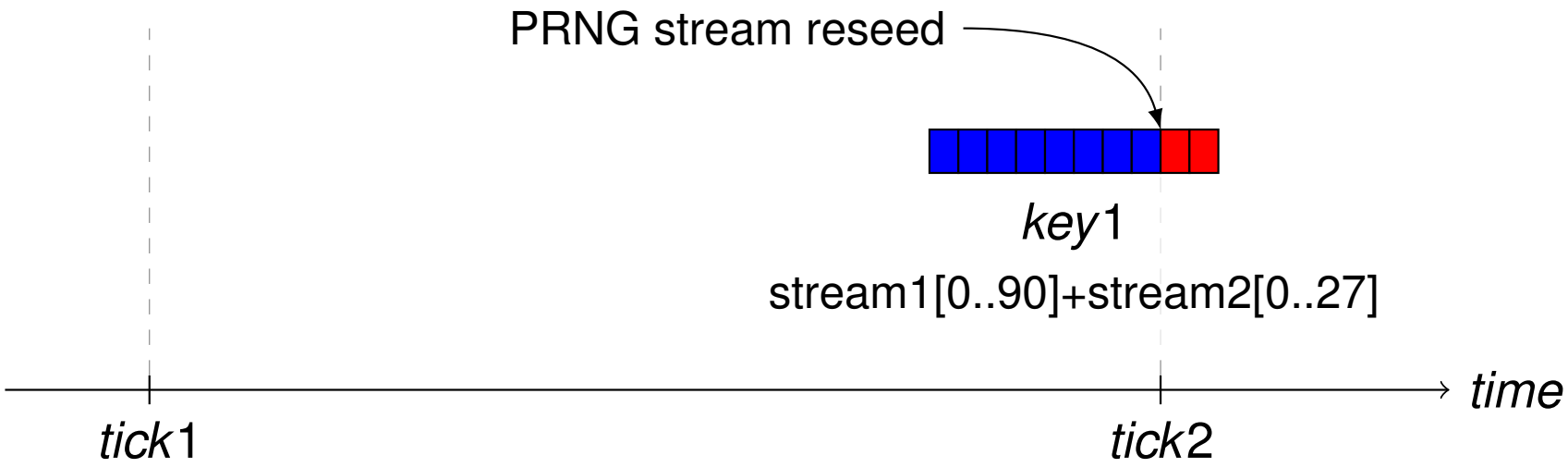
Clé, cas 1



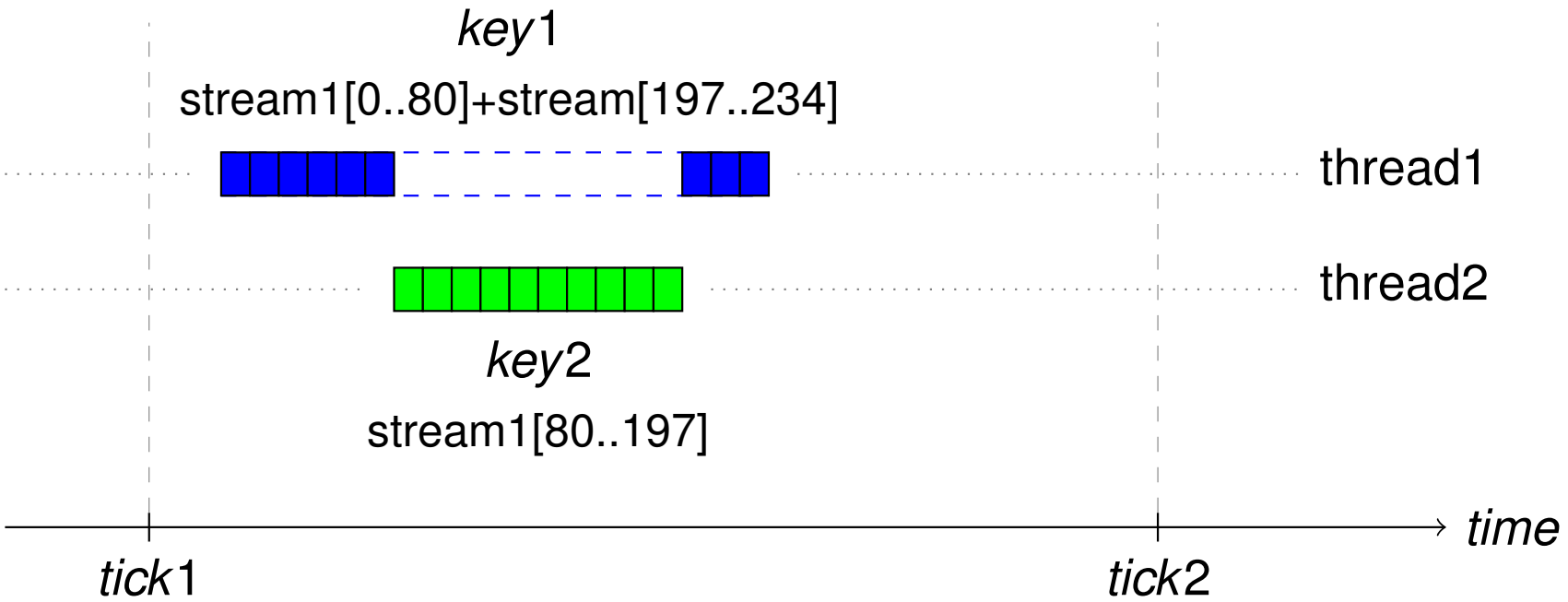
Clé, cas 2


unk
key1
key2
stream1[27..144]
stream1[144..261]
time
tick1
tick2

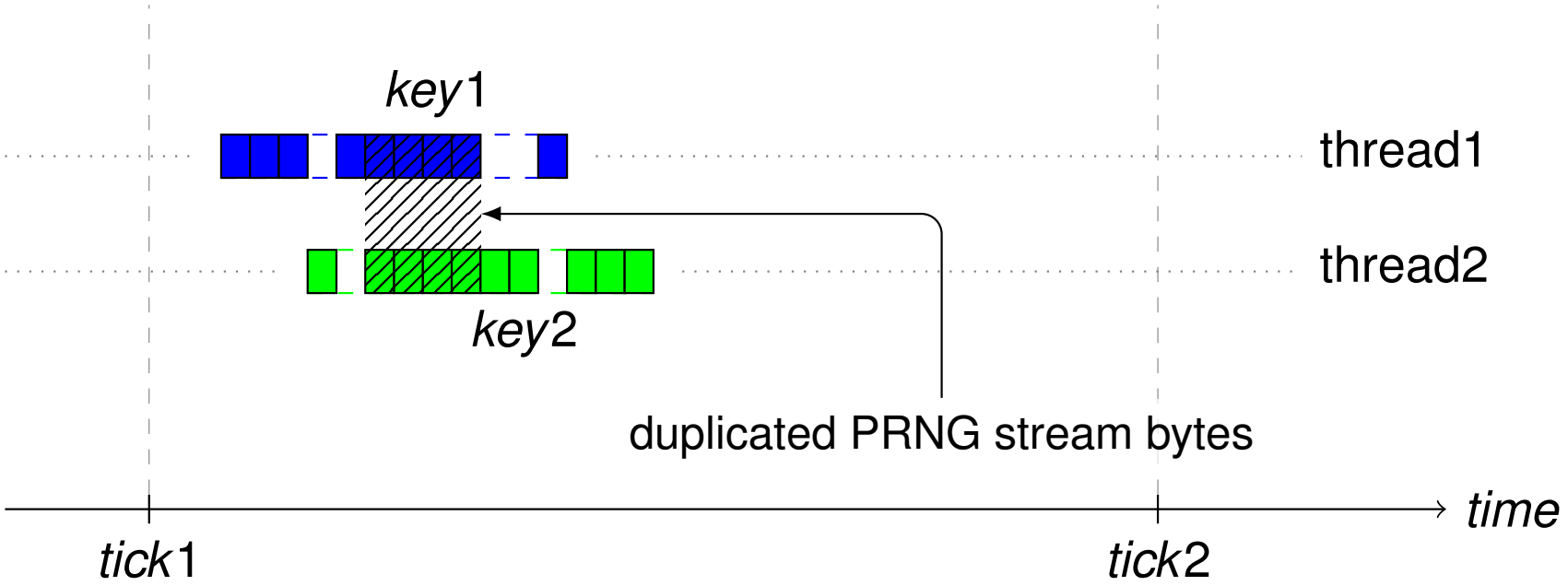
Clé, cas 3


PRNG stream reseed
key1
stream1[0..90]+stream2[0..27]
time
tick1
tick2

Clé, cas 4


key1
stream1[0..80]+stream[197..234]
thread1
thread2
key2
stream1[80..197]
time
tick1
tick2

Clé, cas 5


key1
thread1
thread2
key2
duplicated PRNG stream bytes
time
tick1
tick2

Plan

Reconnaissance d'un fichier déchiffré

- Calcul d'entropie du header
  - Générique
  - Seuil arbitraire
  - Pas forcément adapté à tous les fichiers
- Utilisation de quelques magic
  - En fonction des fichiers fournis

Benchmark

- Tester 2^{32} ticks, 1 clé, 1 fichier : 12h
  - À diviser par le nombre de CPU disponibles
- Augmentation linéaire avec le nombre de
  - ticks
  - clés
  - fichiers

Plan

Limiter les paramètres

- tick == temps
- 1h = 0x0036_0000 (49j = 0xFFFF_FFFF)
- Identifier rapidement 1 valeur
- Élargir à partir de là

Limiter les paramètres

- Éliminer rapidement les fichiers faciles à trouver
  - clés simples
- Tester moins de clés par tick
  - Utiliser l'information existante
  - Adapter les recherches

En pratique

- Un binaire natif (C, OpenMP)
- Un script qui le pilote
- Metadata des clés disponibles
- Enregistrement des actions
  - Évite de dupliquer le travail

En pratique

while (1) {

- Pour chaque tick identifié, cherche le maximum de clés
- Recherche des ticks proches
- Recherche de clés complexes sur les ticks proches
- Sélection de 200 fichiers aléatoires
  - Recherche exhaustive 2^{32} ticks, 1 clé

}

Problèmes

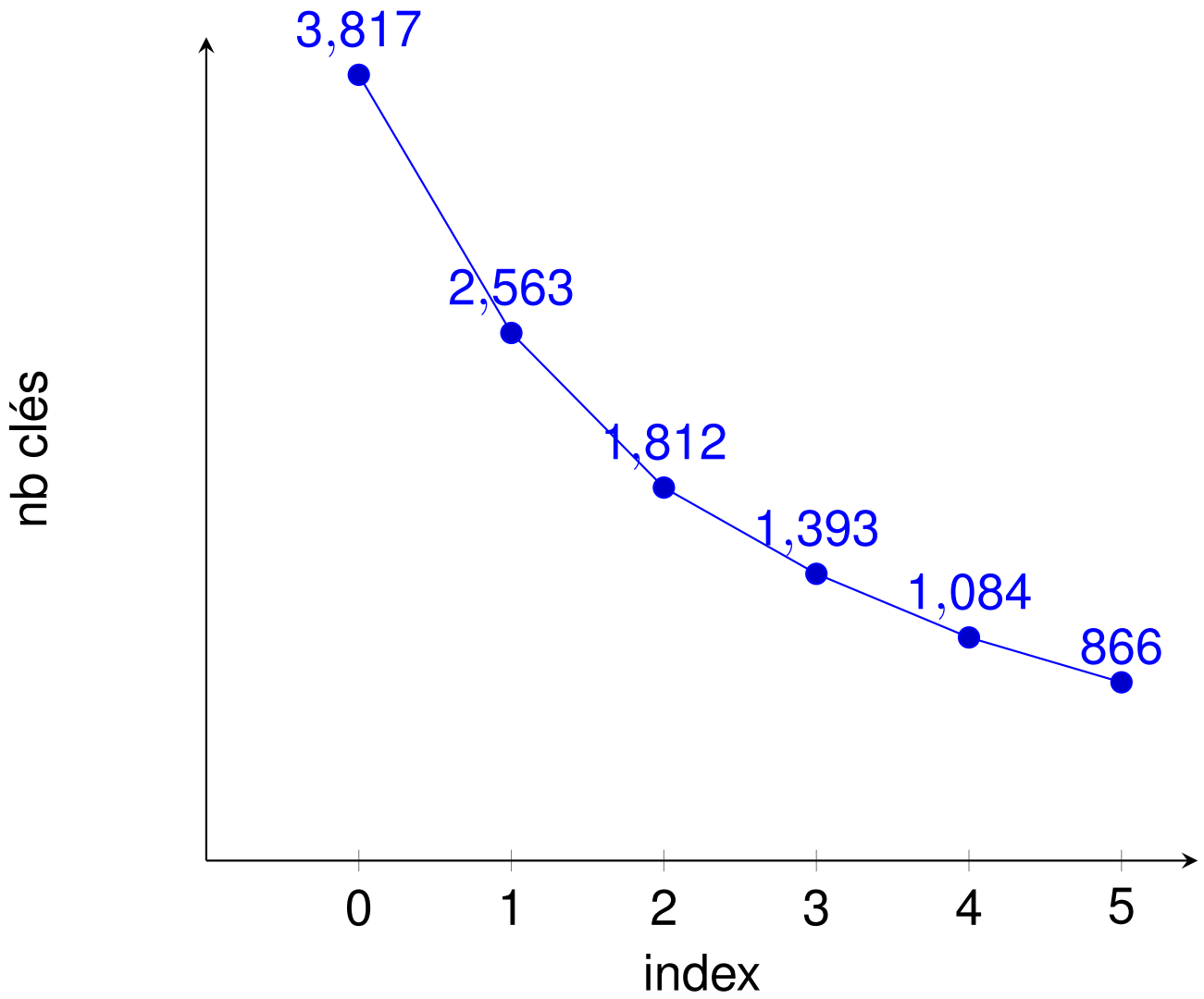
- Petits fichiers (<256)
  - Entropie non fiable
  - Algo custom
    - Compte octets nuls
    - Full ASCII
- Fichiers à forte entropie
  - Test spécifique (file ext)
  - Augmenter le seuil d'entropie en ciblant les clés

Plan

Batch 1

- 16849 fichiers chiffrés, 16056 restaurés (95.3%)
- 4124 valeurs de ticks
- Recherche initiale 2 semaines, rejouable en 8h
- Distribution des clés
  - cas 1 : 92.6% des fichiers
  - cas 1 index 0 : 22.7%
  - cas 2 : 2.7%
  - autres : 7 (0.04%)
  - pas trouvé : 793 (4.7%)
  - 1 tick avec 37 clés

Distribution par index



Batch 2

- 231513 fichiers chiffrés, 207256 restaurés (89.5%)
- 32718 valeurs de ticks
- Distribution des clés
  - cas 1 : 85.8% des fichiers
  - cas 1 index 0 : 30000 (13.7%)
  - cas 2 : 8500 (3.7%)
  - autres : 52
  - pas trouvé : 25000 (10.5%)
  - 1 tick avec 55 clés
  - 750 ticks avec 30+ clés chacun

Conclusion

- Ça reste un cas rare

Conclusion

- Faites des backups (offline) !