Introduction à la rétro-ingénierie logicielle

5d54626cc43ba7ce625903ffa39c049c

Groupe de sécurité de l'information, École normale supérieure

2 octobre 2019

(Rappels) Risques cyber : CIA

Trois grandes catégories de risques aux systèmes d'informations :

(Rappels) Risques cyber : CIA

Trois grandes catégories de risques aux systèmes d'informations :

- **C**onfidentialité : l'information n'est accessible qu'aux personnes autorisées.
- **I**ntégrité : l'information n'est pas modifiée par des actions non autorisées.
- Disponibilité (**A**vailability) : l'information est accessible dans les conditions attendues.

(Rappels) Les modèles d'attaquants

Type	Taille	Ciblage	Technicité	Exemples
Attaquant nul ($\perp$)	0-1	aléatoire	nulle	dégât des eaux, panne, crevaison de pneu, salarié qui trébuche sur un câble.
Attaquant faible	1-3	faible	attaques déjà connues et faciles à réaliser	arnaque nigériane, script kiddie, self-xss
Attaquant fort	1-50	précis	attaque organisée, planifiée, capacité de long terme	Hacking Team, Carbanak
Attaquant absolu ($\top$)	illimitée	unique	absolue : usage de zero-day, puissance de calcul illimitée, cryptologie avancée, agents infiltrés, chantage, cambriolage, "salarié" qui "trébuche" sur un câble	Agences gouvernementales

(Rappels) Les modèles d'attaquants

Type	Taille	Ciblage	Technicité	Exemples
Attaquant nul ($\perp$)	0-1	aléatoire	nulle	dégât des eaux, panne, crevaison de pneu, salarié qui trébuche sur un câble.
Attaquant faible	1-3	faible	attaques déjà connues et faciles à réaliser	arnaque nigériane, script kiddie, self-xss
Attaquant fort	1-50	précis	attaque organisée, planifiée, capacité de long terme	Hacking Team, Carbanak
Attaquant absolu ($\top$)	illimitée	unique	absolue : usage de zero-day, puissance de calcul illimitée, cryptologie avancée, agents infiltrés, chantage, cambriolage, "salarié" qui "trébuche" sur un câble	Agences gouvernementales

D'un coup de pelle, une mamie coupe l'Internet de toute l'Arménie

Par LEXPRESS.fr ,
publié le 08/04/2011 à 11:30

Une mamie récolteuse de cuivre a accidentellement sectionné un cable de fibre optique, déconnectant ainsi tout le pays.

Le 28 mars, les internautes arméniens appuient désespérément sur la touche F5 de leur clavier. Sans succès. Sur les écrans télés, des images de

NEWSLETTER L'EXPRESS

Recevez le meilleur de L'Express sélectionné par la rédaction

Votre adresse e-mail

Ok

(Rappels) Le modèle d'attaquant

Type	Taille	Ciblage	Technicité	Exemples
Attaquant nul ($\perp$)	0-1	aléatoire	nulle	dégât des eaux, panne, crevaison de pneu, salarié qui trébuche sur un câble.
Attaquant faible	1-3	faible	attaques déjà connues et faciles à réaliser	arnaque nigériane script kiddie, self-xss

(Rappels) Le modèle d'attaquant

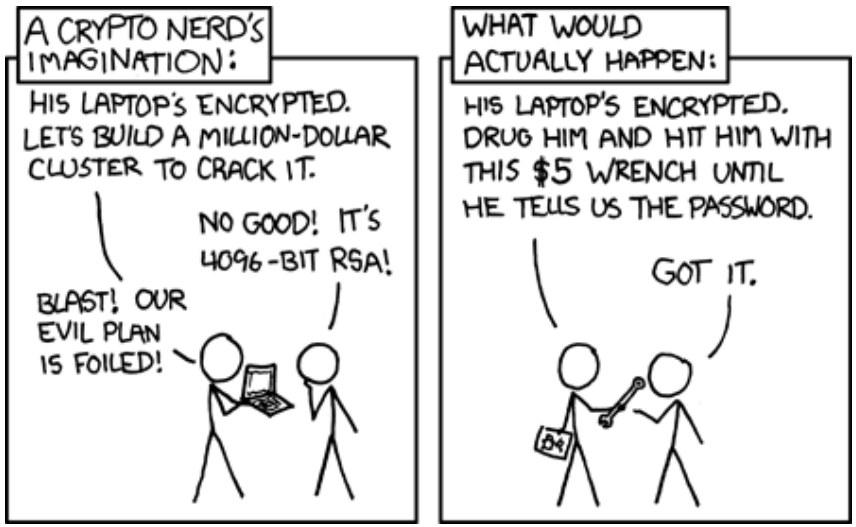
Type	Taille	Ciblage	Technicité	Exemples
Attaquant nul ($\perp$)	0-1	aléatoire	nulle	dégât des eaux, panne, crevaison de pneu, salarié qui trébuche sur un câble.
Attaquant faible	1-3	faible	attaques déjà connues et faciles à réaliser	arnaque nigériane script kiddie, self-xss
Attaquant fort	1-50	précis	attaque organisée, planifiée capacité de long terme	Hacking Team Carbanak

(Rappels) Le modèle d'attaquant

Type	Taille	Ciblage	Technicité	Exemples
Attaquant nul ($\bot$)	0-1	aléatoire	nulle	dégât des eaux, panne, crevaison de pneu, salarié qui trébuche sur un câble.
Attaquant faible	1-3	faible	attaques déjà connues et faciles à réaliser	arnaque nigériane script kiddie, self-xss
Attaquant fort	1-50	précis	attaque organisée, planifiée capacité de long terme	Hacking Team Carbanak
Attaquant absolu ($\top$)	illimitée	unique	absolue : usage de zero-day, puissance de calcul illimitée, cryptologie avancée, agents infiltrés, chantage, cambriolage, "salarié" qui "trébuche" sur un câble	Agences gouvernementales

(Rappels) Le modèle d'attaquant

Type	Taille	Ciblage	Technicité	Exemples
Attaquant nul ($\bot$)	0-1	aléatoire	nulle	dégât des eaux, panne, crevaison de pneu, salarié qui trébuche sur un câble.
Attaquant faible	1-3	faible	attaques déjà connues et faciles à réaliser	arnaque nigériane script kiddie, self-xss
Attaquant fort	1-50	précis	attaque organisée, planifiée capacité de long terme	Hacking Team Carbanak
Attaquant absolu ($\top$)	illimitée	unique	absolue : usage de zero-day, puissance de calcul illimitée, cryptologie avancée, agents infiltrés, chantage, cambriolage, "salarié" qui "trébuche" sur un câble	Agences gouvernementales



Plan de ce cours

- 2 octobre 2019
  - Introduction au reverse
  - Compilation et assembleur
  - Rétroingénierie statique
- 9 octobre 2019
  - Rétroingénierie dynamique
  - Rétroingénierie en pratique
  - Les bases de l'exploitation
  - Travaux pratiques
- 16 octobre 2019
  - Techniques avancées de reverse
  - Introduction aux méthodes formelles
  - Examen

La rétro-ingénierie, c'est quoi ?

- Spécification
- Conception
- Implémentation
- Comportement

La rétro-ingénierie, c'est quoi ?

- Spécification
- Conception
- Implémentation
- Comportement

Definition (Rétro-ingénierie, n.f.)

Ensemble des opérations d'analyse d'un logiciel ou d'un matériel destinées à retrouver le processus de sa conception et de sa fabrication, ainsi que les modalités de son fonctionnement.

Pourquoi faire de la rétro-ingénierie ?

Plusieurs finalités :

- Pédagogique : comprendre un objet pour l'utiliser correctement.

Pourquoi faire de la rétro-ingénierie ?

Plusieurs finalités :

- Pédagogique : comprendre un objet pour l'utiliser correctement.
- Interopérabilité : interfacer un produit inconnu avec un autre.

Pourquoi faire de la rétro-ingénierie ?

Plusieurs finalités :

- Pédagogique : comprendre un objet pour l'utiliser correctement.
- Interopérabilité : interfacer un produit inconnu avec un autre.
- Forensique : extraire des informations sur les utilisateurs ou les concepteurs d'un produit.

Pourquoi faire de la rétro-ingénierie ?

Plusieurs finalités :

- Pédagogique : comprendre un objet pour l'utiliser correctement.
- Interopérabilité : interfacer un produit inconnu avec un autre.
- Forensique : extraire des informations sur les utilisateurs ou les concepteurs d'un produit.
- Certification : garantir la conformité d'un produit à des normes.

Pourquoi faire de la rétro-ingénierie ?

Plusieurs finalités :

- Pédagogique : comprendre un objet pour l'utiliser correctement.
- Interopérabilité : interfacer un produit inconnu avec un autre.
- Forensique : extraire des informations sur les utilisateurs ou les concepteurs d'un produit.
- Certification : garantir la conformité d'un produit à des normes.
- Espionnage industriel : comprendre un objet en vue de le copier.

Pourquoi faire de la rétro-ingénierie ?

Plusieurs finalités :

- Pédagogique : comprendre un objet pour l'utiliser correctement.
- Interopérabilité : interfacer un produit inconnu avec un autre.
- Forensique : extraire des informations sur les utilisateurs ou les concepteurs d'un produit.
- Certification : garantir la conformité d'un produit à des normes.
- Espionnage industriel : comprendre un objet en vue de le copier.
- Recherche de vulnérabilités : comprendre un processus en vue de le détourner de son fonctionnement nominal.

Exemple de rétro-ingénierie

- Pédagogique : qui a déjà ouvert un pc ?
- Interopérabilité : samba, SIP.
- Forensique : Apple vs FBI (San Bernardino).
- Certification : DieselGate.
- Espionnage industriel : 5G, RQ170, Arduino nano.
- Recherche de vulnérabilités : Enigma.

Cadre légal

Article L122-6-1

III. La personne ayant le droit d'utiliser le logiciel peut sans l'autorisation de l'auteur **observer**, **étudier** ou **tester le fonctionnement** ou la **sécurité** de ce logiciel afin de déterminer les idées et principes qui sont à la base de n'importe quel élément du logiciel [...].
IV. La reproduction du code du logiciel ou la traduction de la forme de ce code n'est pas soumise à l'autorisation de l'auteur lorsque la reproduction ou la traduction au sens du 1° ou du 2° de l'article L. 122-6 est indispensable pour obtenir les informations nécessaires à l'**interopérabilité d'un logiciel créé de façon indépendante** avec d'autres logiciels, sous réserve que soient réunies les conditions suivantes :

1. Ces actes sont accomplis par la personne ayant le droit d'utiliser un exemplaire du logiciel ou pour son compte par une personne habilitée à cette fin ;
2. Les informations nécessaires à l'interopérabilité n'ont pas déjà été rendues facilement et rapidement accessibles aux personnes mentionnées au 1° ci-dessus ;
3. Et ces actes sont limités aux parties du logiciel d'origine nécessaires à cette interopérabilité.

Cadre légal

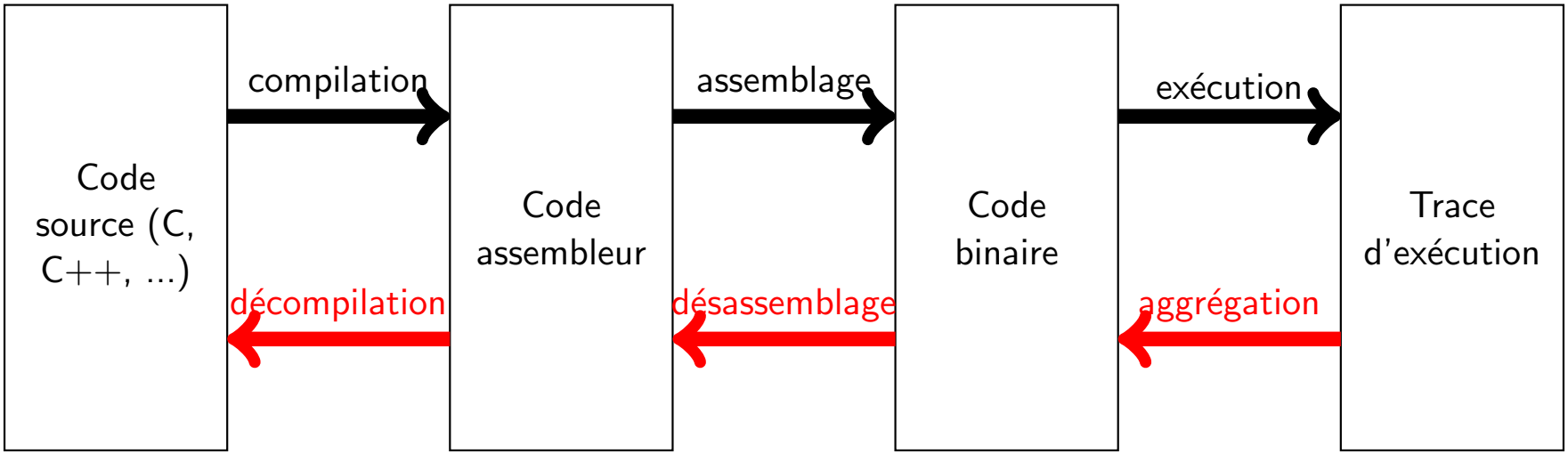
Mais des restrictions sur le résultat :

Suite de l'article L122-6-1

Les informations ainsi obtenues ne peuvent être :

1. Ni utilisées à des fins autres que la réalisation de l'interopérabilité du logiciel créé de façon indépendante ;
2. Ni communiquées à des tiers sauf si cela est nécessaire à l'interopérabilité du logiciel créé de façon indépendante ;
3. Ni utilisées pour la mise au point, la production ou la commercialisation d'un logiciel dont l'expression est substantiellement similaire ou pour tout autre acte portant atteinte au droit d'auteur.

Vue d'ensemble



* hors cas particulier pour les langages interprétés (Java, Python, .NET, ...)

** ne prend pas en compte l'édition de liens.

Différentes techniques

- **Boite noire** : accès uniquement au comportement du programme (relations entrées-sorties)
  Boite blanche : accès aux traces/binaires/...

Différentes techniques

- **Boite noire** : accès uniquement au comportement du programme (relations entrées-sorties)
  Boite blanche : accès aux traces/binaires/...
- **Analyse statique** : indépendante d'une exécution particulière
  Analyse dynamique : étude durant une ou plusieurs exécutions du programme.
  Analyse concolique : mélange des deux

Différentes techniques

- **Boite noire** : accès uniquement au comportement du programme (relations entrées-sorties)
  Boite blanche : accès aux traces/binaires/...
- **Analyse statique** : indépendante d'une exécution particulière
  Analyse dynamique : étude durant une ou plusieurs exécutions du programme.
  Analyse concolique : mélange des deux
- **Analyse passive** : observation extérieure du programme (pas d'interaction).
  Analyse active : possibiltié de choisir les entrées et sorties du programme.

Que cherche-t-on dans un programme ?

- Protocoles (USB, IP, ...)
- Données utilisateur (nom de domaine, numéro de série, ...)
- Eléments cryptographiques (clé, S-boîte)
- Algorithmes implémentés
- Effets de bords, connectivité réseau
- Bibliothèques utilisées

Quelques outils connus

- Les hardcores : `objdump`, `objcopy`, `strings`, `readelf`, `gdb`, `qemu`, `strace`, `ltrace`, `binwalk`, `hexedit`
- Les classiques : `IDA Pro` (et `Hexrays`), `ghidra`, `Radare2`, `OllyDBG`, `PEiD`, `CFF Explorer`, `bindiff`, `valgrind`
- Les spécialisés : `miasm`, `Pin`, `angr`, `Apktool`, profilers divers et variés, `afl` (et autres fuzzers).

Quelques outils connus (demo)

objdump

```
objdump -d a.out | less
```

strings

```
string a.out | less
```

readelf

```
readelf -a a.out | less
```

Quelques outils connus (demo)

ghidra

`https://ghidra-sre.org/`

La rétro-ingénierie vue comme abstraction

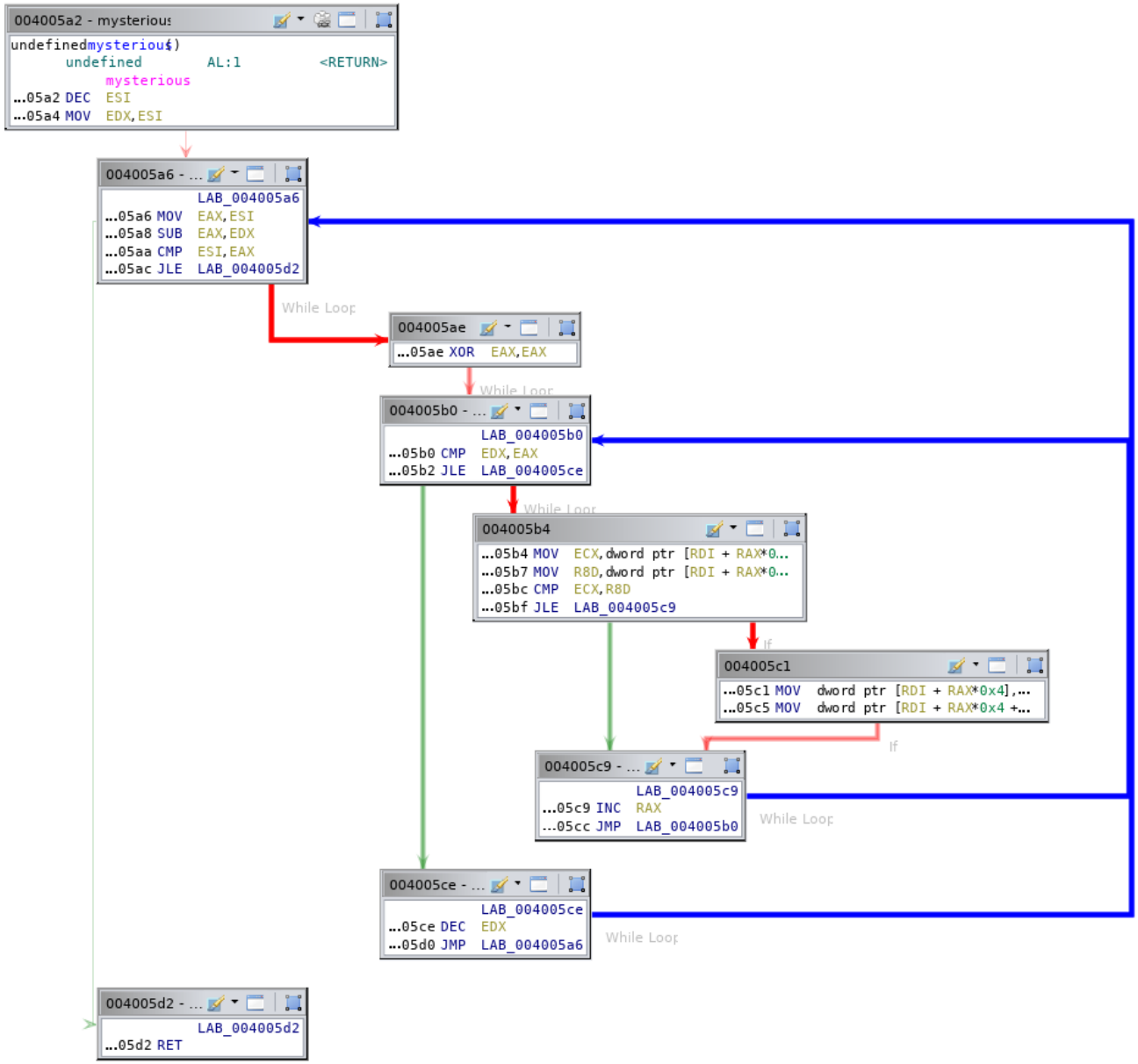
- Un programme peut aussi être vu comme concrétisation d'une idée. Le but du reverse se voit alors comme la remontée successive dans les couches d'abstractions afin de retrouver l'idée originale.
- Le résultat doit donc être une information générale, minimalement dépendante du programme étudié.

La rétro-ingénierie vue comme abstraction : exemple

```
00005a0: eb8e ffce 89f2 89f0 29d0 39c6 7e24 31c0  ........).9.~$1.
00005b0: 39c2 7e1a 8b0c 8744 8b44 8704 4439 c17e  9.~....D.D..D9.~
00005c0: 0844 8904 8789 4c87 0448 ffc0 ebe2 ffca  .D....L..H......
00005d0: ebd4 c341 5449 89fc 5589 f553 31db 39dd  ...ATI..U..S1.9.
```

```
0000000004005a2 <mysterious>:
  4005a2:       ff ce                   dec    %esi
  4005a4:       89 f2                   mov    %esi,%edx
  4005a6:       89 f0                   mov    %esi,%eax
  4005a8:       29 d0                   sub    %edx,%eax
  4005aa:       39 c6                   cmp    %eax,%esi
  4005ac:       7e 24                   jle    4005d2 <mysterious+0x30>
  4005ae:       31 c0                   xor    %eax,%eax
  4005b0:       39 c2                   cmp    %eax,%edx
  4005b2:       7e 1a                   jle    4005ce <mysterious+0x2c>
  4005b4:       8b 0c 87                mov    (%rdi,%rax,4),%ecx
  4005b7:       44 8b 44 87 04          mov    0x4(%rdi,%rax,4),%r8d
  4005bc:       44 39 c1                cmp    %r8d,%ecx
  4005bf:       7e 08                   jle    4005c9 <mysterious+0x27>
  4005c1:       44 89 04 87             mov    %r8d,(%rdi,%rax,4)
  4005c5:       89 4c 87 04             mov    %ecx,0x4(%rdi,%rax,4)
  4005c9:       48 ff c0                inc    %rax
  4005cc:       eb e2                   jmp    4005b0 <mysterious+0xe>
  4005ce:       ff ca                   dec    %edx
  4005d0:       eb d4                   jmp    4005a6 <mysterious+0x4>
  4005d2:       c3                      retq
```

La rétro-ingénierie vue comme abstraction : exemple



La rétro-ingénierie vue comme abstraction : exemple

```
1
2  void mysterious(int *arr,int n)
3
4  {
5    int tmp;
6    long j;
7    int i;
8
9    n = n + -1;
10   i = n;
11   while (n - i < n) {
12     j = 0;
13     while ((int)j < i) {
14       tmp = arr[j];
15       if (arr[j + 1] < tmp) {
16         arr[j] = arr[j + 1];
17         arr[j + 1] = tmp;
18       }
19       j = j + 1;
20     }
21     i = i + -1;
22   }
23   return;
24 }
25
```

La rétro-ingénierie vue comme abstraction : exemple

C'est donc un tri à bulle.

```
/*@
requires \valid(arr + (0 ..  n-1));
requires 0 < n;
assigns a[0 ..  n-1];
ensures \forall integer i, j; 0 <= i <= j < n ==> arr[i] <=
arr[j];
*/
```

Ce qu'on survolera dans ce cours sans s'attarder

- Rétro-ingénierie matérielle
- Cryptanalyse
- Compilation de langages non impératifs
- Jeu d'instruction assembleur spécialisés
- Protections d'espace exécutable (ASLR, NX, Canaries, ...)

Pause

Des questions ?

Pour ceux qui veulent s'amuser en reversant :
`https://microcorruption.com/`