

Anti-patrons en CTF

Une analyse empirique de l'échec collaboratif

Jean Privat

Hubert Hackin'

2026-09-22 SECUQAM

Capture The Flag (CTF)

Compétition où les participants

- Recherche de l'information secrète, le drapeau
- « Vend » le drapeau au système de score contre des points
- Celui qui a le plus de points gagne

Ya pas de métho. infaillible

- Reconnaissance initiale
- Tester le chemin “normal”, reproduire localement
- Chercher les trucs douteux, les chemins d'exploitation
- Faire le travail

Write-up

- Rapport écrit qui détaille la solution d'une épreuve
- Ce que l'auteur a trouvé suffisamment intéressant pour être rédigé
- Technique, pas forcément long ou exhaustif, rarement littéraire

Hubert Hackin'

- Ref. Bâtiment A de l'UQAM
- Participe au NorthSec depuis 2016
 - Top 3 depuis 2018
- Nos write-ups sont réputés décalés
 - <https://hubert.hackin.ca>

Anti-patrons en CTF

Une analyse empirique de l'échec collaboratif

- Corpus : write-ups de Hubert Hackin'
- Terrain : CTF, fatigue et outils hostiles
- Objet : erreurs commises, évitées ou sublimées

Corpus

- Write-ups 2020 à 2026
- Web, reverse, crypto, misc, hardware, CI
- Succès, échecs et solutions post-CTF

Limite

Les write-ups racontent mieux les victoires que les heures perdues

10. Implémenter avant de comprendre

The Cabal (rev, NSEC21)

- Un programme COBOL obfusqué
- Réimplémentation en python
- Boucle pour trouver le bon antécédent par force brute

002-MAGIC-OP.

MOVE 1 TO BIT-VAL.

MOVE ZERO TO RETRN.

IF ARG1 IS NOT EQUAL TO ZERO OR ARG2 IS NOT EQUAL TO ZERO

PERFORM 003-MAGIC-OP-SUB

UNTIL ARG1 IS EQUAL TO ZERO AND ARG2 IS EQUAL TO ZERO.

003-MAGIC-OP-SUB.

DIVIDE ARG1 BY 2 GIVING QTN1.

COMPUTE $RSD1 = ARG1 - QTN1 * 2$.

DIVIDE ARG2 BY 2 GIVING QTN2.

COMPUTE $RSD2 = ARG2 - QTN2 * 2$.

IF RSD1 IS NOT EQUAL TO RSD2 THEN

ADD BIT-VAL TO RETRN

END-IF.

MULTIPLY BIT-VAL BY 2 GIVING BIT-VAL.

MOVE QTN1 TO ARG1.

MOVE QTN2 TO ARG2.

- La fonction COBOL compare les bits de deux nombres
- Chaque fois que les bits diffèrent,
- il ajoute la puissance de deux correspondante au résultat

- La fonction COBOL compare les bits de deux nombres
- Chaque fois que les bits diffèrent,
- il ajoute la puissance de deux correspondante au résultat

Eh mais c'est XOR

- La fonction COBOL compare les bits de deux nombres
- Chaque fois que les bits diffèrent,
- il ajoute la puissance de deux correspondante au résultat

Eh mais c'est XOR

- En pratique, la force brute fonctionne très bien ici
- Donc seulement 10e place des anti-patrons

9. Chercher seulement ce que l'on connaît

Math Homework 3 (NSEC20)

```
def answer
  @answer ||= @array.send(@type)
end
```

- Un serveur en Ruby
 - Possibilité limité d'importer du code
- Besoin de trouver
 - une méthode @type sans argument à envoyer à @array
 - → On parle de « gadget »
- Le participant
 - Utilise sa connaissance de la lib standard de Ruby
 - Fouille la doc pendant longtemps
 - Mais ne trouve aucune fonction utilisable

9. Chercher seulement ce que l'on connaît

Math Homework 3 (NSEC20)

```
def answer
  @answer ||= @array.send(@type)
end
```

- Un serveur en Ruby
 - Possibilité limité d'importer du code
- Besoin de trouver
 - une méthode @type sans argument à envoyer à @array
 - → On parle de « gadget »
- Le participant
 - Utilise sa connaissance de la lib standard de Ruby
 - Fouille la doc pendant longtemps
 - Mais ne trouve aucune fonction utilisable
- Un vrai recon aurait donné de quoi
 - @type="constants"; @array=Module;
 - → La vraie liste des vraies classes où chercher des gadgets (Gem)

8. L'ordinateur a toujours raison

Eemoji (pwn, 0CTF'20)

- Un binaire avec des emojis
- On utilise décompilateur de Ghidra
- Bizarreries subtiles dans le C
- Qu'est-ce que cette diablerie?

```

void FUN_00100b91(void)
{
    wchar_t *pwVar1;

    if (DAT_00302030 == '\0') {
        fputws(L" \xfe0f\n",stdout);
        /* WARNING: Subroutine does not return */
        abort();
    }
    memset(DAT_00302038,0x41,0x1000);
    fgetws(DAT_00302038,0x82,stdin);
    pwVar1 = DAT_00302038;
    *(undefined8 *)((long)DAT_00302038 + 0x202) = 0x9090909090909090;
    *(undefined8 *)((long)pwVar1 + 0x20a) = 0x9090909090909090;
    *(undefined8 *)((long)pwVar1 + 0x212) = 0x48ffffffe00b28d48;
    *(undefined8 *)((long)pwVar1 + 0x21a) = 0xc74800000001c7c7;
    *(undefined8 *)((long)pwVar1 + 0x222) = 0xc0c74800000026c2;
    *(undefined4 *)((long)pwVar1 + 0x22a) = 1;
    *(undefined2 *)((long)pwVar1 + 0x22e) = 0x50f;
    *(undefined *)((long)pwVar1 + 0x23c) = 0xc3;
    memset(DAT_00302038 + 0x40,0x41,0x100);
    (*(code *) (DAT_00302038 + 0x80))();
    return;
}

```

- Ghidra était bogué, les offsets sont pas bons
- Une analyse de l'assembleur montre la supercherie

```

00100c1f 48 89 50 10 MOV    qword ptr [RAX + 0x10],RDX
00100c23 48 89 48 18 MOV    qword ptr [RAX + 0x18],RCX
00100c27 48 8b 15 MOV    RDX,qword ptr [DAT_00100fe0]           = C0C74800000026C2h
b2 03 00 00
00100c2e 48 89 50 20 MOV    qword ptr [RAX + 0x20],RDX
00100c32 8b 15 b0 MOV    EDX,dword ptr [DAT_00100fe8]           = 00000001h
03 00 00
00100c38 89 50 28 MOV    dword ptr [RAX + 0x28],EDX
00100c3b 0f b7 15 MOVZX  EDX,word ptr [DAT_00100fec]           = 050Fh
aa 03 00 00
00100c42 66 89 50 2c MOV    word ptr [RAX + 0x2c],DX
00100c46 0f b6 15 MOVZX  EDX,byte ptr [DAT_00100fee]           = C3h
a1 03 00 00
00100c4d 8b 50 2e MOV    byte ptr [RAX + 0x2e],DL
00100c50 48 8b 05 MOV    RAX,qword ptr [DAT_00302038]
e1 13 20 00
00100c57 48 05 00 ADD     RAX,0x100
01 00 00
00100c5d ba 00 01 MOV    EDX,0x100
00 00
00100c62 be 41 00 MOV    ESI,0x41
00 00
00100c67 48 89 c7 MOV    RDI,RAX
00100c6a e8 91 fc CALL   memset
ff ff void * memset(void *
00100c6f 48 8b 05 MOV    RAX,qword ptr [DAT_00302038]
c2 13 20 00
00100c76 48 05 00 ADD     RAX,0x200
02 00 00
00100c7c 48 89 c2 MOV    RDX,RAX
00100c7f b8 00 00 MOV    EAX,0x0
00 00
00100c84 ff d2 CALL   RDX
00100c86 90 NOP
00100c87 5d POP    RBP
00100c88 c3 RET

```

```

memset(DAT_00302038 + 0x100, 'A',0x100);
(*(code *) (DAT_00302038 + 0x200))();
return;

```

7. Ne pas remettre en question une déduction

Rare Metal Sequencer 2/2 (rev, NSEC21)

- Une ROM super-nintendo
 - Demande d'entrer la bonne séquence de touches
- Rétro-ingénierie
 - Gestion des contrôleurs
 - Masque de bits
 - Arithmétique
- Il faut appuyer sur les boutons dans le bon ordre
 - Partie 1: OK
 - Partie 2: Il faut appuyer sur 9 boutons exactement en même temps

7. Ne pas remettre en question une déduction

Rare Metal Sequencer 2/2 (rev, NSEC21)

- Une ROM super-nintendo
 - Demande d'entrer la bonne séquence de touches
- Rétro-ingénierie
 - Gestion des contrôleurs
 - Masque de bits
 - Arithmétique
- Il faut appuyer sur les boutons dans le bon ordre
 - Partie 1: OK
 - Partie 2: Il faut appuyer sur 9 boutons exactement en même temps
- Trop difficile!
 - Les participants continuent la rétroingénierie
 - Déterminent comment le flag est généré

7. Ne pas remettre en question une déduction

Rare Metal Sequencer 2/2 (rev, NSEC21)

- Une ROM super-nintendo
 - Demande d'entrer la bonne séquence de touches
- Rétro-ingénierie
 - Gestion des contrôleurs
 - Masque de bits
 - Arithmétique
- Il faut appuyer sur les boutons dans le bon ordre
 - Partie 1: OK
 - Partie 2: Il faut appuyer sur 9 boutons exactement en même temps
- Trop difficile!
 - Les participants continuent la rétroingénierie
 - Déterminent comment le flag est généré
- En fait, la logique était mal comprise
 - Il n'y avait pas 9 boutons en même temps
 - Pas besoin de rétroingénierie compliquée

6. Laisser le concepteur nous perturber

QR-Code 1/2 (rev, NSEC20)

- Un binaire OCaml
- Demande un Code QR



- Comment entrer un code QR?
- Quels caractères spéciaux?
- Quel codage?

- Rétroingénierie?
 - 1344 fonctions
 - Bien compliqué à analyser

- Rétroingénierie?
 - 1344 fonctions
 - Bien compliqué à analyser

```
my_encoded_input = 2LL * my_input[v7] + 1;
item = get_item_at_index(*(hardcoded_array + 0x18), a6);
v16 = *(item + 0x18);
if ( *get_item_at_index(*(hardcoded_array_ + 0x18), *(item + 0x18)) != 3LL || (my_encoded_input - 0xE2) <= 3 )
```

- Concepteur est un coquin
 - $0xE2 \gg 1 == 0x71$ // 'q'
 - $0xE4 \gg 1 == 0x72$ // 'r'
 - → Le programme s'attend à une chaîne sur [qr]*

5. Optimiser une mauvaise approche

QR-Code 2/2 (rev, NSEC20)

- 278 automates (DFA) à résoudre
 - Trouver la chaîne sur $[qr]^*$ reconnue par les 278

5. Optimiser une mauvaise approche

QR-Code 2/2 (rev, NSEC20)

- 278 automates (DFA) à résoudre
 - Trouver la chaîne sur $[qr]^*$ reconnue par les 278
- proto en Python → ça explose

5. Optimiser une mauvaise approche

QR-Code 2/2 (rev, NSEC20)

- 278 automates (DFA) à résoudre
 - Trouver la chaîne sur $[qr]^*$ reconnue par les 278
- proto en Python $\rightarrow$ ça explose
- Mémoiser $\rightarrow$ ça explose

5. Optimiser une mauvaise approche

QR-Code 2/2 (rev, NSEC20)

- 278 automates (DFA) à résoudre
 - Trouver la chaîne sur $[qr]^*$ reconnue par les 278
- proto en Python → ça explose
- Mémoiser → ça explose
- Recoder en Nit → ça explose

5. Optimiser une mauvaise approche

QR-Code 2/2 (rev, NSEC20)

- 278 automates (DFA) à résoudre
 - Trouver la chaîne sur $[qr]^*$ reconnue par les 278
- proto en Python $\rightarrow$ ça explose
- Mémoiser $\rightarrow$ ça explose
- Recoder en Nit $\rightarrow$ ça explose
- Simplifier préalablement les automates $\rightarrow$ ça explose

5. Optimiser une mauvaise approche

QR-Code 2/2 (rev, NSEC20)

- 278 automates (DFA) à résoudre
 - Trouver la chaîne sur `[qr]*` reconnue par les 278
- proto en Python → ça explose
- Mémoiser → ça explose
- Recoder en Nit → ça explose
- Simplifier préalablement les automates → ça explose
- Perdu, plus de temps :(

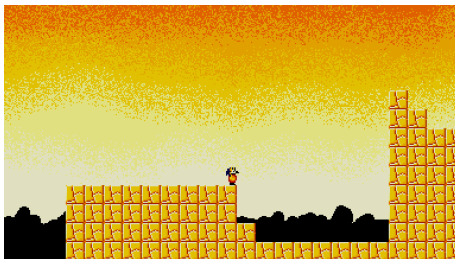
5. Optimiser une mauvaise approche

QR-Code 2/2 (rev, NSEC20)

- 278 automates (DFA) à résoudre
 - Trouver la chaîne sur `[qr]*` reconnue par les 278
- proto en Python → ça explose
- Mémoiser → ça explose
- Recoder en Nit → ça explose
- Simplifier préalablement les automates → ça explose
- Perdu, plus de temps :(
- Les automates ont une structure particulière
 - recoder les automates en z3 → ça passe

4. Beaucoup d'énergie au mauvais endroit

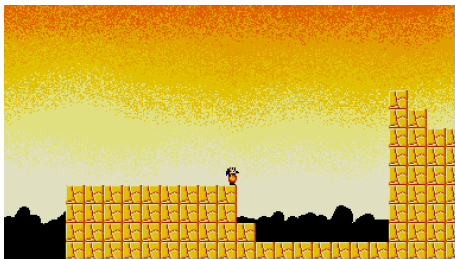
Façade (steg, Athack'24)



- Un jeu de plateforme, impossible à finir
- Tricher pour finir le jeu
 - Long et compliqué → Ne donne rien au final

4. Beaucoup d'énergie au mauvais endroit

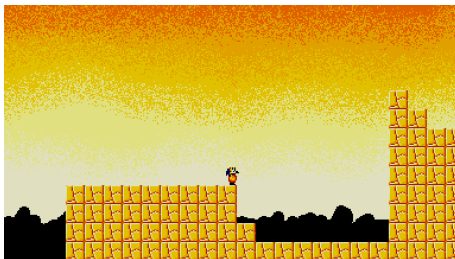
Façade (steg, Athack'24)



- Un jeu de plateforme, impossible à finir
- Tricher pour finir le jeu
 - Long et compliqué → Ne donne rien au final
- Reverse d'un main de 2 700 lignes
 - Long et compliqué → Ne donne rien au final

4. Beaucoup d'énergie au mauvais endroit

Façade (steg, Athack'24)



- Un jeu de plateforme, impossible à finir
- Tricher pour finir le jeu
 - Long et compliqué → Ne donne rien au final
- Reverse d'un main de 2 700 lignes
 - Long et compliqué → Ne donne rien au final
- C'est de la steg!
 - Longue exploration des fichiers ressources → Flag trouvé

3. Ne pas prendre de pause

Façade (steg, Athack'24) (suite)



- « After being up for nearly 26 hours, I even managed to convince myself that I had uncovered a Y in this image (to the great amusement of my teammates). Can you see it? Tell me I'm not alone on this. »

2. Travailler seul

NSEC23 Code Pipeline 1/3

```
$ git ls-files  
.ci/run  
.prettierrc.toml  
main.js
```

```
$ cat .ci/run  
#!/usr/bin/env bash  
set -ex  
echo "linting changes with Prettier..."  
prettier --check .
```

- Un outil est exécuté en CI/CD
- On veut exécuter du code arbitraire sur le serveur
- On bricole pour trouver de quoi
 - et si main.js était un répertoire?

2. Travailler seul

NSEC23 Code Pipeline 1/3

```
$ git ls-files  
.ci/run  
.prettierrc.toml  
main.js
```

```
$ cat .ci/run  
#!/usr/bin/env bash  
set -ex  
echo "linting changes with Prettier..."  
prettier --check .
```

- Un outil est exécuté en CI/CD
- On veut exécuter du code arbitraire sur le serveur
- On bricole pour trouver de quoi
 - et si `main.js` était un répertoire?
 - un lien symbolique vers `/etc/passwd`?

2. Travailler seul

NSEC23 Code Pipeline 1/3

```
$ git ls-files
.ci/run
.prettierrc.toml
main.js
```

```
$ cat .ci/run
#!/usr/bin/env bash
set -ex
echo "linting changes with Prettier..."
prettier --check .
```

- Un outil est exécuté en CI/CD
- On veut exécuter du code arbitraire sur le serveur
- On bricole pour trouver de quoi
 - et si main.js était un répertoire?
 - un lien symbolique vers /etc/passwd?
 - vers proc/self/envIRON?

2. Travailler seul

NSEC23 Code Pipeline 1/3

```
$ git ls-files
.ci/run
.prettierrc.toml
main.js
```

```
$ cat .ci/run
#!/usr/bin/env bash
set -ex
echo "linting changes with Prettier..."
prettier --check .
```

- Un outil est exécuté en CI/CD
- On veut exécuter du code arbitraire sur le serveur
- On bricole pour trouver de quoi
 - et si main.js était un répertoire?
 - un lien symbolique vers /etc/passwd?
 - vers proc/self/envIRON?
 - etc.

2. Travailler seul

NSEC23 Code Pipeline 1/3

```
$ git ls-files
.ci/run
.prettierrc.toml
main.js
```

```
$ cat .ci/run
#!/usr/bin/env bash
set -ex
echo "linting changes with Prettier..."
prettier --check .
```

- Un outil est exécuté en CI/CD
- On veut exécuter du code arbitraire sur le serveur
- On bricole pour trouver de quoi
 - et si main.js était un répertoire?
 - un lien symbolique vers /etc/passwd?
 - vers proc/self/envIRON?
 - etc.
- « prettier is a real tool »

1. Ne pas RTFM

Rules (recon, NSEC21)

- 2018, 3e participation de l'équipe Hubert Hackin'
- Égalité entre les deux premières équipes
- Improvisation d'un jeu décisif pour déterminer le gagnant final
- Hubert Hackin' finit second :(

1. Ne pas RTFM

Rules (recon, NSEC21)

- 2018, 3e participation de l'équipe Hubert Hackin'
- Égalité entre les deux premières équipes
- Improvisation d'un jeu décisif pour déterminer le gagnant final
- Hubert Hackin' finit second :(
- Personne dans l'équipe n'avait soumis le flag gratuit des règles

1. Ne pas RTFM

Rules (recon, NSEC21)

- 2018, 3e participation de l'équipe Hubert Hackin'
- Égalité entre les deux premières équipes
- Improvisation d'un jeu décisif pour déterminer le gagnant final
- Hubert Hackin' finit second :(
- Personne dans l'équipe n'avait soumis le flag gratuit des règles
 - RTFM

Conclusion

- Vérifier les hypothèses bon marché
- Communiquer plus tôt
- Revenir aux données brutes
- Dormir
- RTFM

Dans le doute: documenter, partager, puis aller dormir.