Hacking de jeux vidéo:

Créer des challs et les casser!

Lucas PARSY

1

Bases du hack de jeux

You know the rules and so do I

Serious Games

- **1 milliard** de joueurs en ligne.
  37% admettent avoir déjà triché

Serious Games

- 1 milliard de joueurs en ligne.
  37% admettent avoir déjà triché

- **76 milliards $ microtransactions en 2023**

Serious Games

- 1 milliard de joueurs en ligne.
  37% admettent avoir déjà triché

- 76 milliards $ microtransactions en 2023

- **Compétitions avec cashprizes énormes**

Différentes méthodes de hack de jeux

Extraction d'assets

ILSpy

Décompilation

Ghidra

Obfuscation

RenderDoc

GPU API

CE Autoassembler

Hooking

Injection de DLL

Frida

Unity Explorer

Bypass Anti-cheat

Code Filter

Détection de valeurs

Scan mémoire

Debugging

Cheat Engine

Vérifications serveur

Pwn multijoueur

Interception réseau

Burp

Chiffrement SSL

Edit registre

Slowdown système

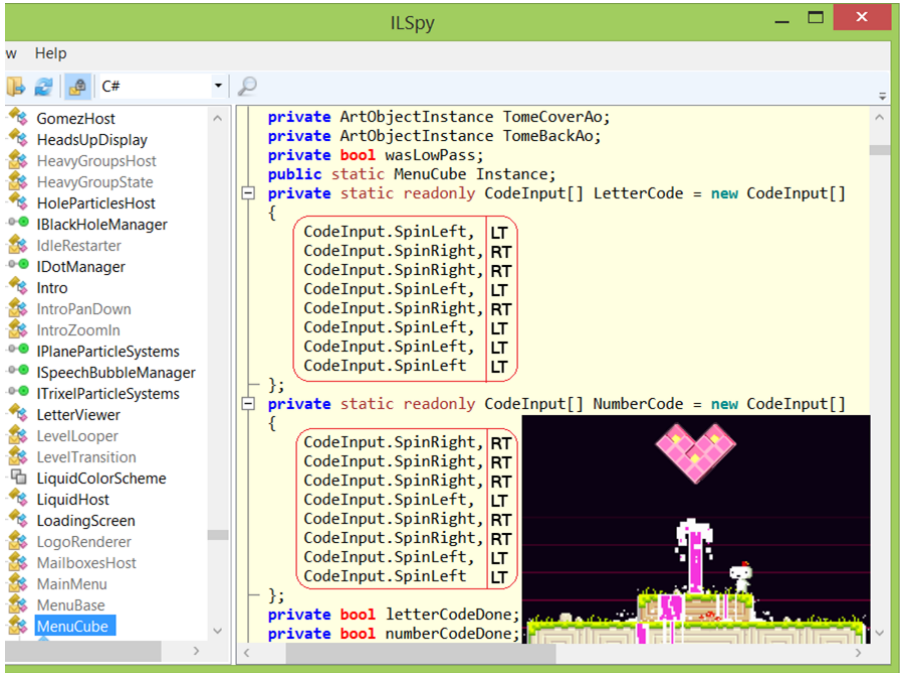
modifs système

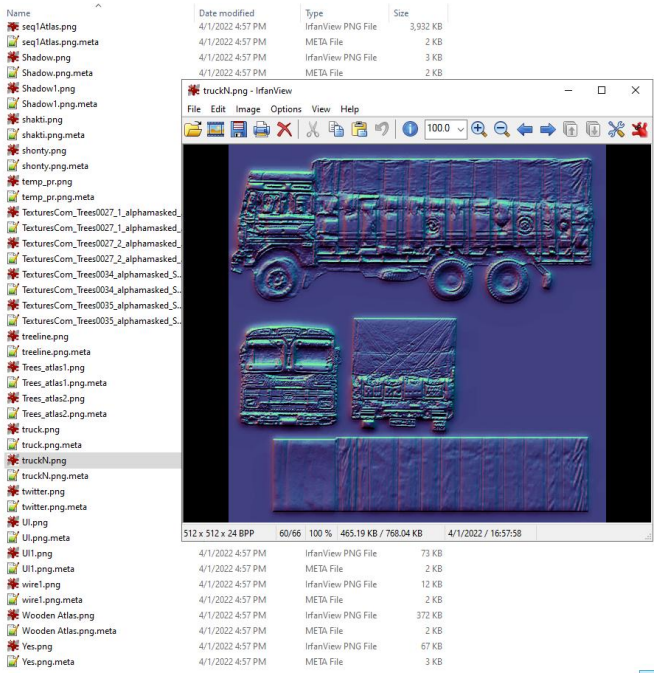
Changement horloge

Edit fichiers sauvegarde

Différentes méthodes de hack de jeux

Unpack/Décompilation de code source





Différentes méthodes de hack de jeux

Unpack/Décompilation de code source

Différentes méthodes de hack de jeux

Hooking de fonctions

process

FRIDA

```
session = frida.attach("solitaire.exe")
script = session.create_script("""
   Interceptor.attach(ptr(ADRESSE_FONCTION),
   {
      onEnter(args) {
         args[0] = ptr("1337");
      }
   });
""")
```

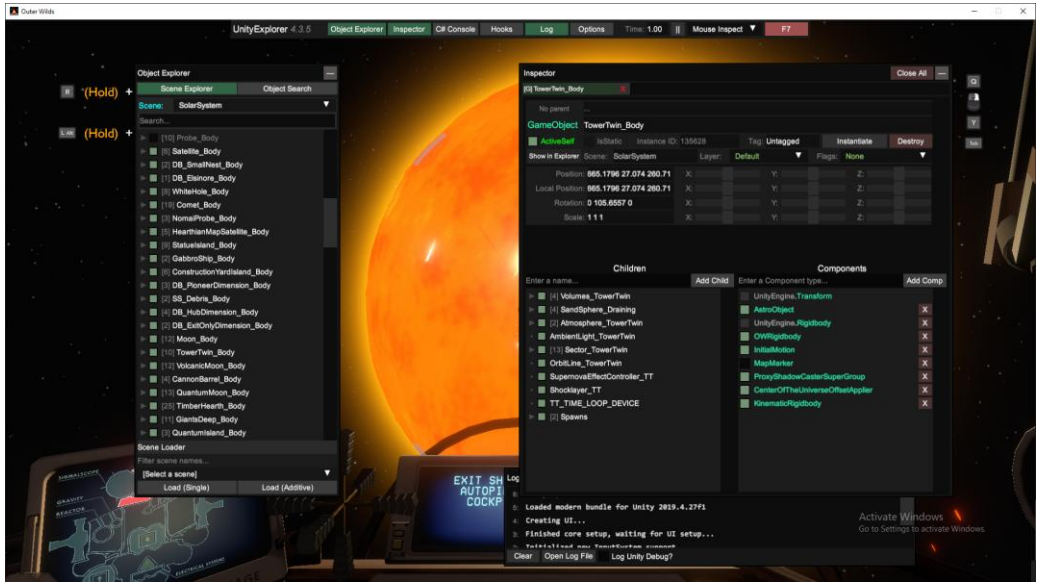
Différentes méthodes de hack de jeux

Hooking de fonctions

Moteur de jeu

BepInEx / MelonLoader

process



Différentes méthodes de hack de jeux

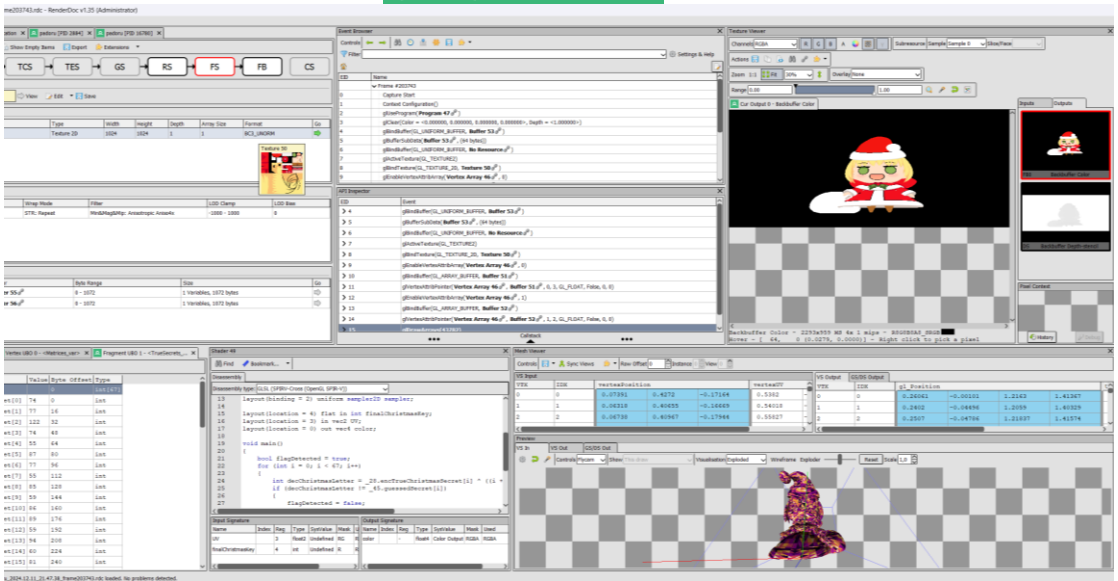
Hooking de fonctions

Moteur de jeu

Render GPU

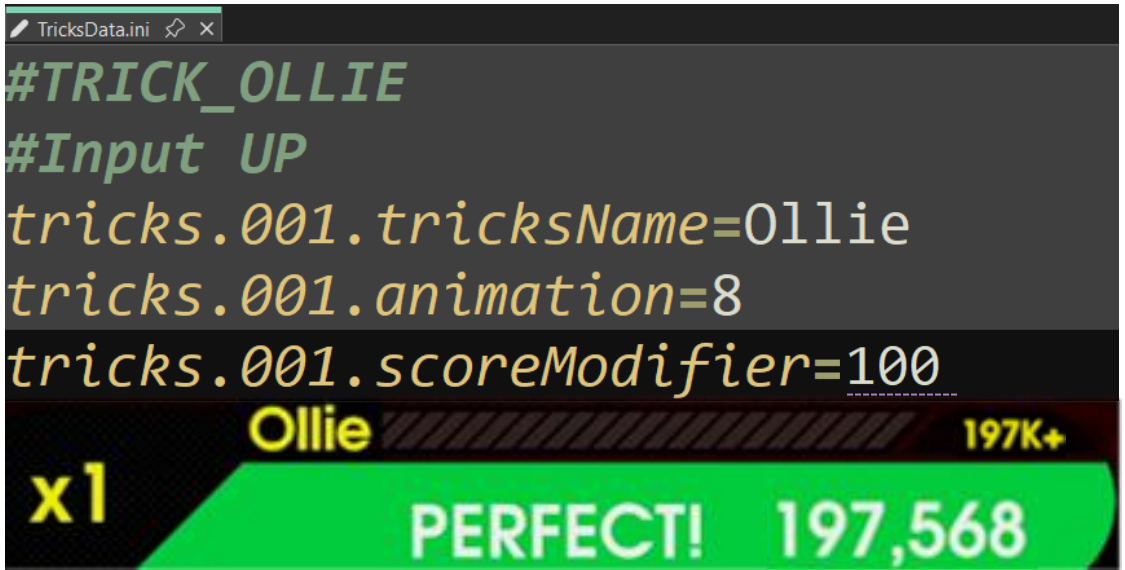
process

RenderDoc



Différentes méthodes de hack de jeux

Edit de fichiers de sauvegarde et config

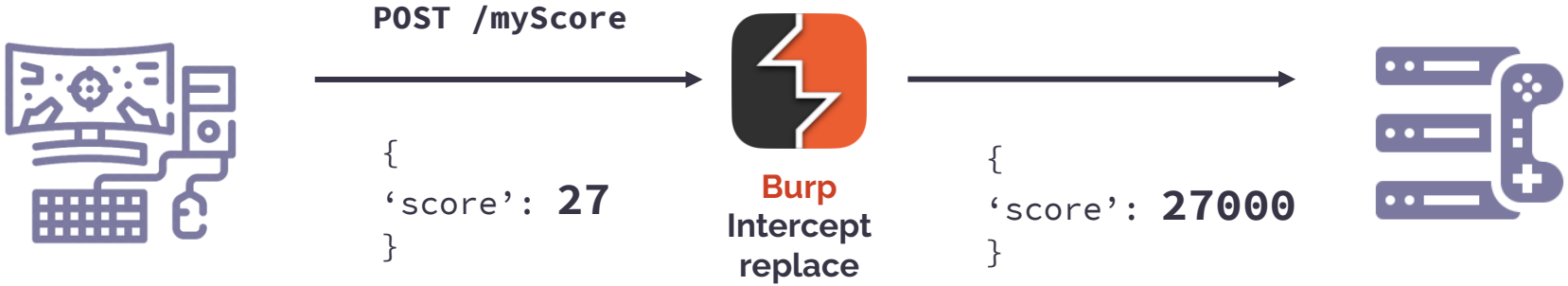


Procmon64

Trouve les fichiers/ Clés registres accédés par un process

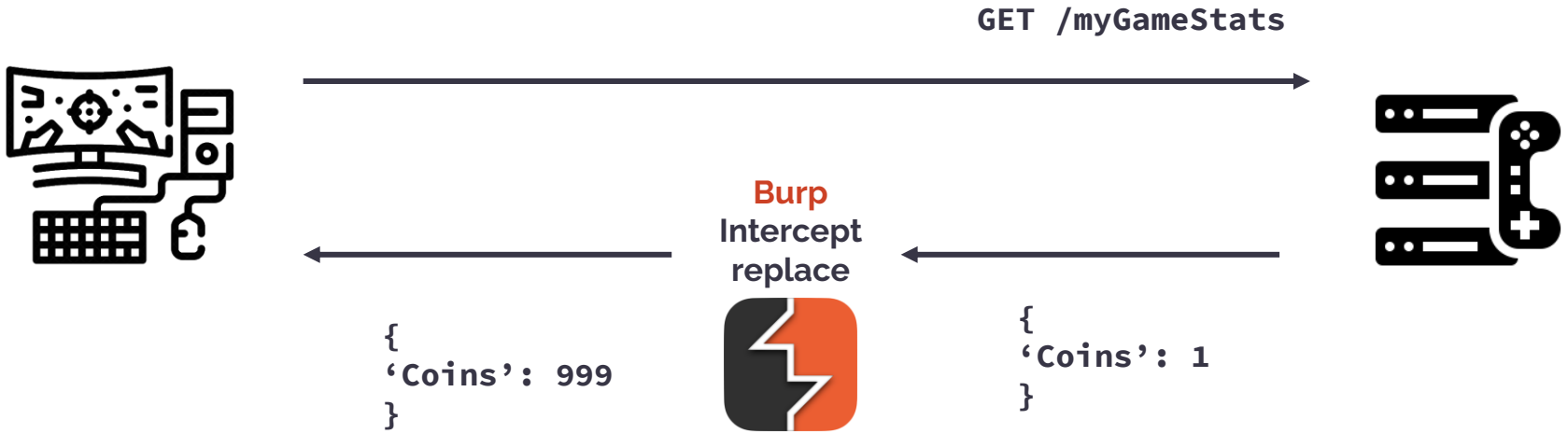
Différentes méthodes de hack de jeux

Interception/modification de paquets réseau



Differentes méthodes de hack de jeux

Interception/modification de packets réseau



Différentes méthodes de hack de jeux

- Interception/modification de paquets réseau

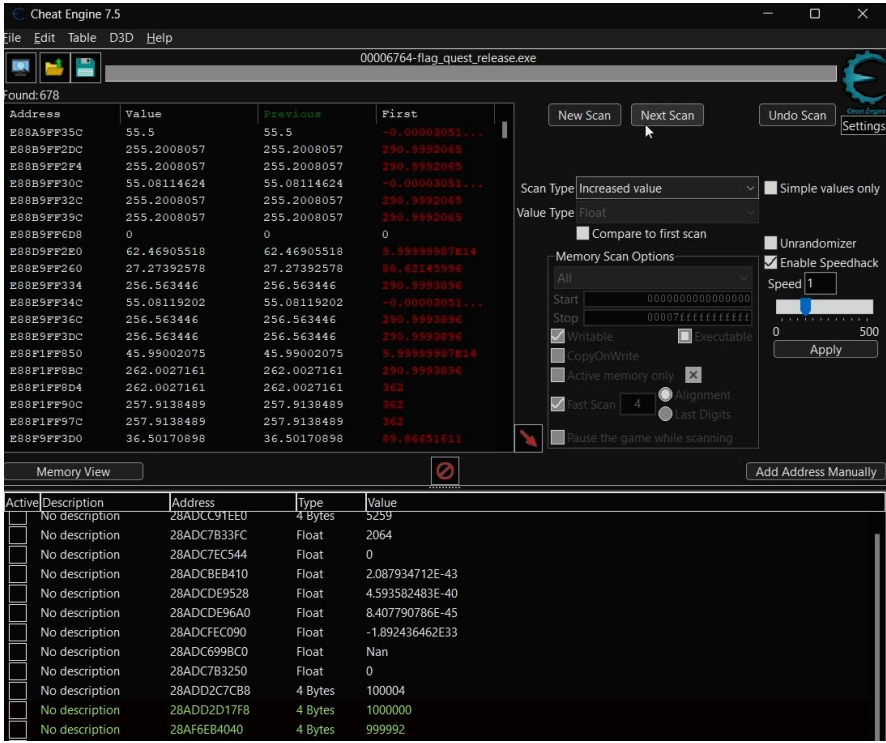
2

Hacker des jeux avec CheatEngine

"Ça se prononce Aine jean"

Cheat Engine

- Scanner mémoire et debugger



Cheat Engine

- Scanner mémoire et debugger
- **scripting AutoAssembly et LUA**

```
Hook :
retGetGamePlayers_o:
readmem( retGetGamePlayers, 6 )
mov [LocalPlayer],rax
mov rcx, [rax+30]
test rcx,rcx
je short @f
    mov [OakPlayerController],rcx
    mov rcx, [rcx+488]
    test rcx, rcx
        je short @f
        mov rcx, [rax+30]
        mov rcx, [rcx+1988]
        test rcx,rcx
        je short @f
            mov [OakDeveloperPerks],rcx
            test byte ptr [rcx+C8],40
            jne short @f
                or byte ptr [rcx+C8],40

@@:
jmp retGetGamePlayers+6
```

Cheat Engine

- Scanner mémoire et debugger
- **scripting AutoAssembly et LUA**

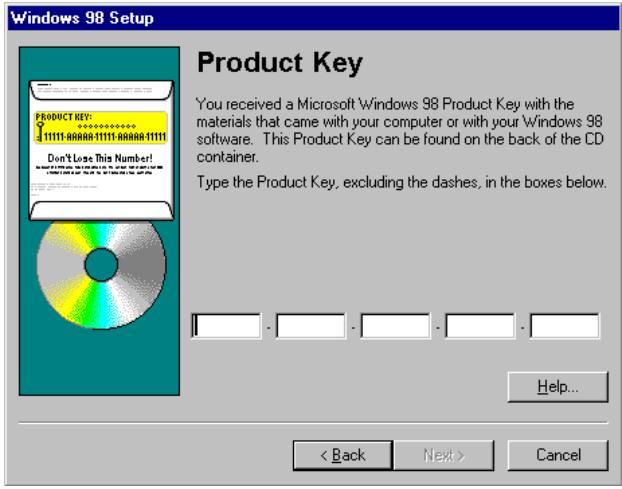
```
function AOBScanAA(script, symbol)
  local success,disableInfo = autoAssemble(script)
  if not success then return nil, disableInfo end -- disab
  local addr = getAddress(symbol)
  autoAssemble(script, disableInfo) -- disable script and
  return addr, 'success'
end

function AOBScanRegion(bytestr, start, stop)
  local script = ([[
  [ENABLE]
  aobscanregion(luaAOBScanRegionSymbol,%X,%X,%s)
  registersymbol(luaAOBScanRegionSymbol)
  [DISABLE]
  unregistersymbol(luaAOBScanRegionSymbol)
  ]]):format(getAddress(start), getAddress(stop), bytestr)
  return AOBScanAA(script, 'luaAOBScanRegionSymbol')
end

function AOBScanModule(bytestr, module)
  local script = ([[
  [ENABLE]
  aobscanmodule(luaAOBScanModuleSymbol,%s,%s)
  registersymbol(luaAOBScanModuleSymbol)
  [DISABLE]
  unregistersymbol(luaAOBScanModuleSymbol)
  ]]):format(module, bytestr)
  return AOBScanAA(script,'luaAOBScanModuleSymbol')
end
```

Cheat Engine

- Scanner mémoire et debugger
- scripting AutoAssembly et LUA
- **Générateur de GUI ‘trainer’**

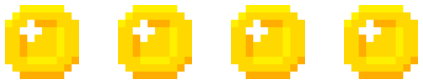
CheatEngine

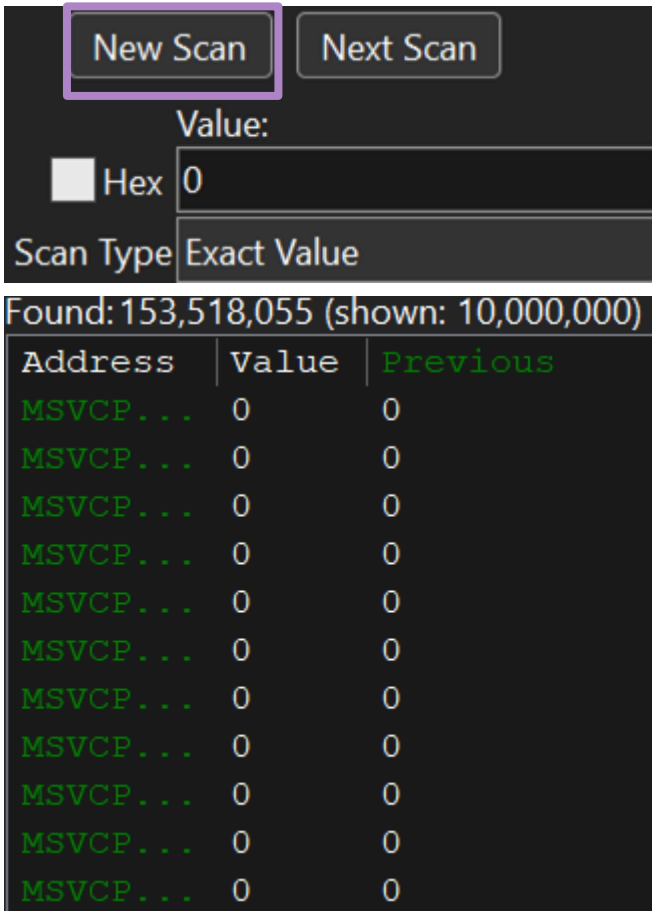
- Scanner mémoire et debugger
- scripting AutoAssembly et LUA
- Générateur de GUI ‘trainer’
- **Pas limité au hacking de jeux**



Trouver des valeurs

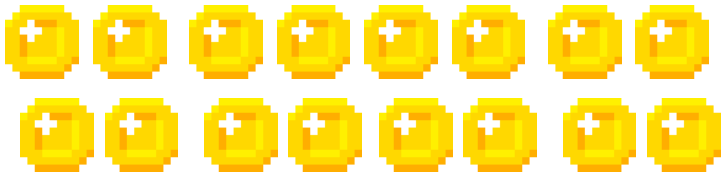
COINS : 0





Trouver des valeurs

COINS : 4



New Scan | Next Scan

Value: 4

Hex

Scan Type: Exact Value

Found: 113

Address	Value	Previous
1318B...	0	4
1318B...	0	4
13193...	4	4
13193...	4	4
13193...	4	4
13193...	4	4
13193...	4	4
15C0A...	4	4

Trouver des valeurs

COINS : 20

New Scan | Next Scan

Value: 20

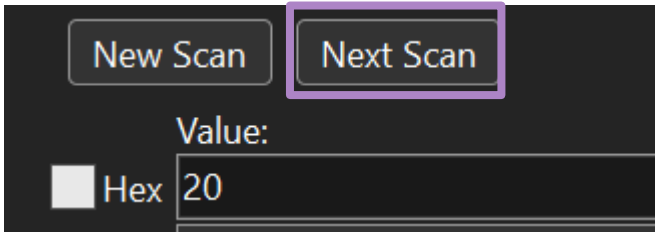
Hex

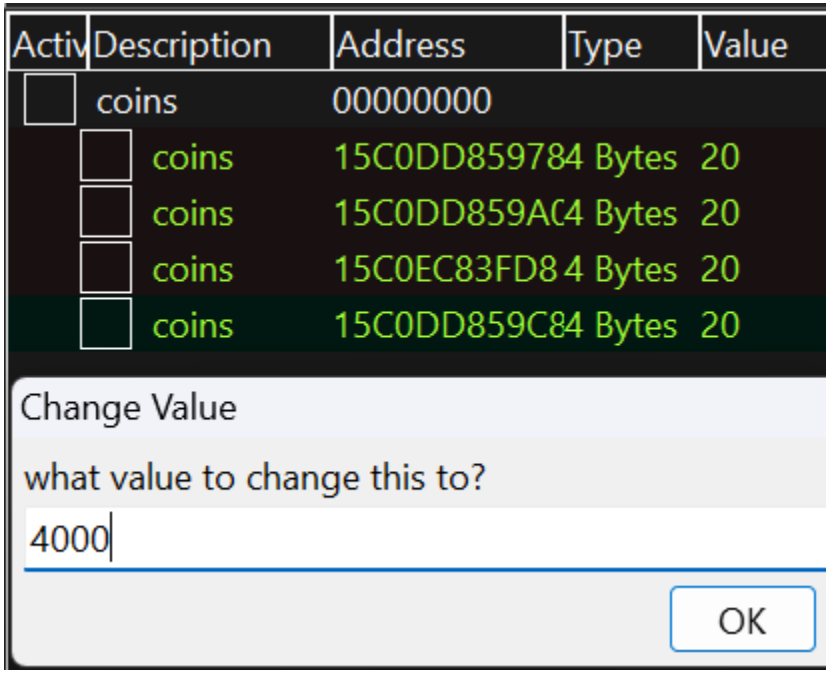
Found:4

Address	Value	Previous
15C0D...	20	20
15C0D...	20	20
15C0D...	20	20
15C0E...	20	20

Trouver des valeurs

COINS : 20





Trouver des valeurs

COINS : 4000

Activ	Description	Address	Type	Value
☐	coins	00000000		
☐	coins	15C0DD85978	4 Bytes	20
☐	coins	15C0DD859AC	4 Bytes	20
☐	coins	15C0EC83FD8	4 Bytes	20
☐	coins	15C0DD859C8	4 Bytes	20

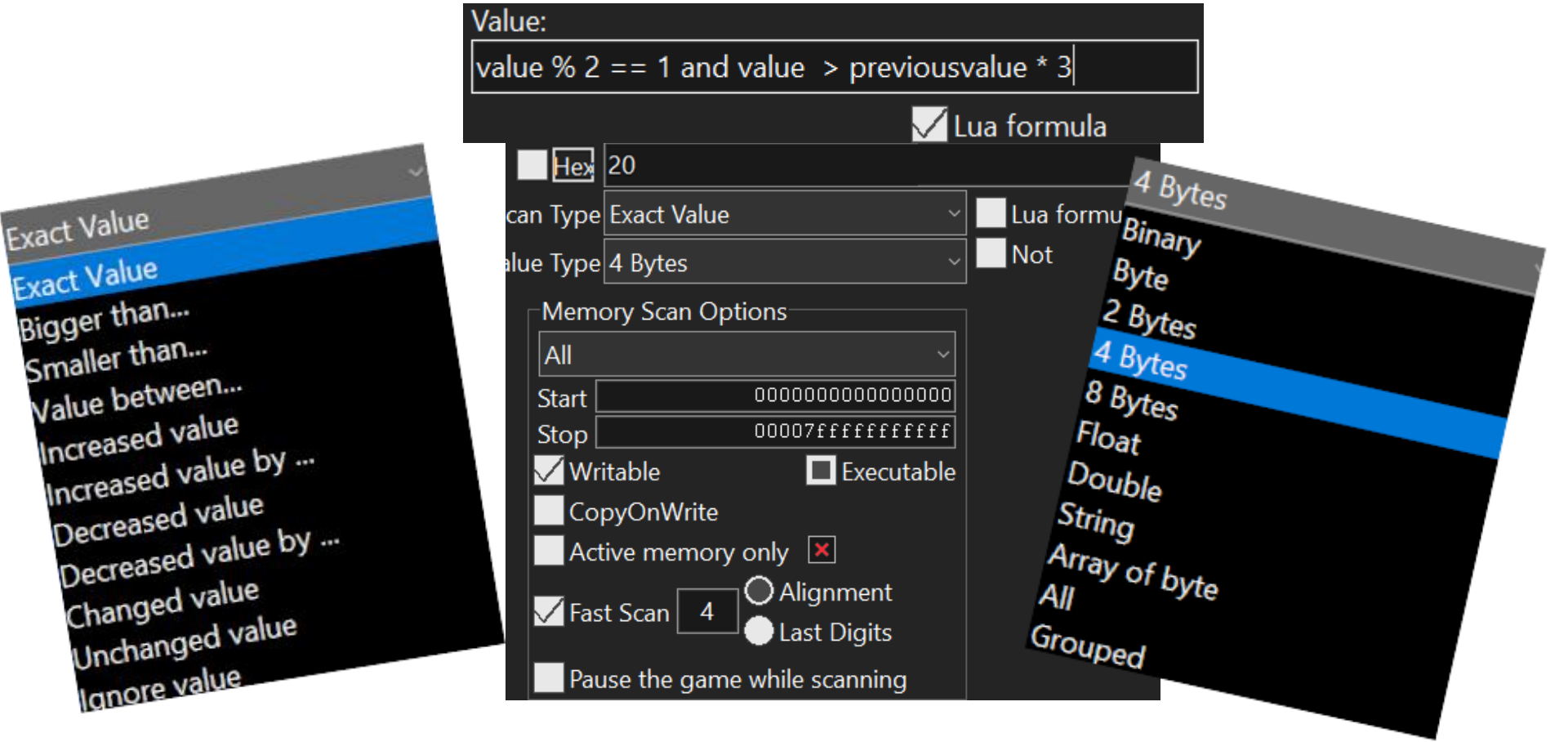
Change Value

what value to change this to?

4000

OK

Trouver des valeurs: features avancées



Memory viewer

Description	Address	Type	Value
health	15C0DD85978	4 Bytes	1337

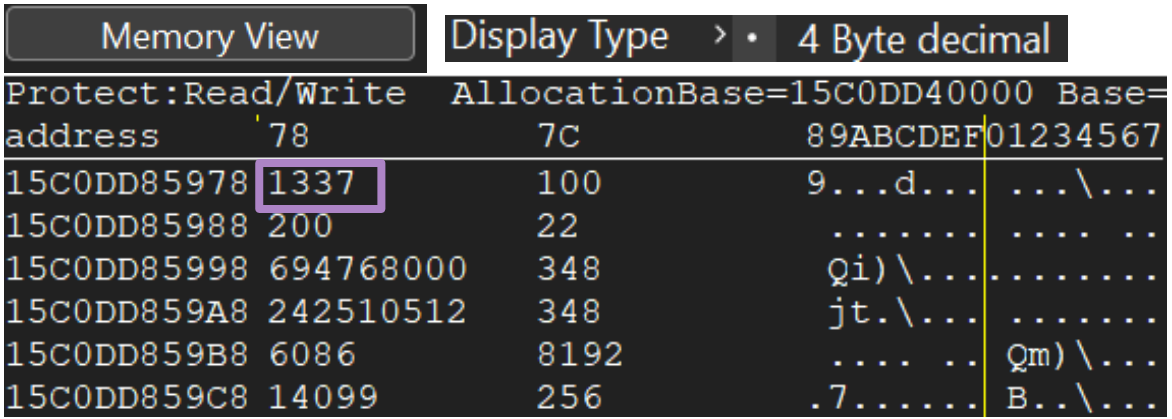
Memory View

```
Protect:Read/Write  AllocationBase=15C0DD40000 Base=
address      80 81 82 83 84 85 86 87 89ABCDEF01234567
15C0DD85978 39 05 00 00 64 00 00 00 9...d...  ...\...
15C0DD85988 C8 00 00 00 16 00 00 00  ........  .... ..
15C0DD85998 80 51 69 29 5C 01 00 00  Qi)\....  ........
15C0DD859A8 B0 6A 74 0E 5C 01 00 00  jt.\....  ........
15C0DD859B8 C6 17 00 00 00 20 00 00  .... ..  Qm)\...
15C0DD859C8 13 37 00 00 00 01 00 00 .7......  B..\...
```

```
struct Player
{
    int health   = 1337;
    int ???      = ???;
    int ???      = ???;
}
```

Memory viewer

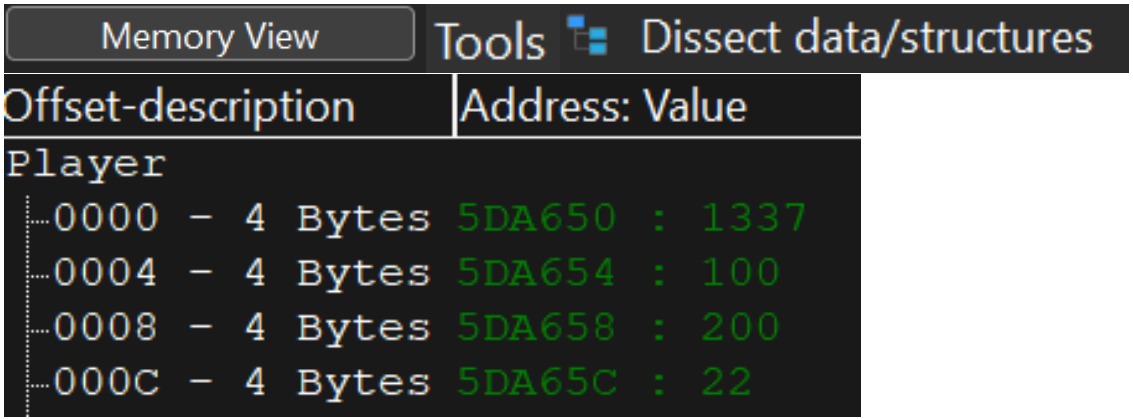
Description	Address	Type	Value
health	15C0DD85978	4 Bytes	1337




```
struct Player
{
    int health   = 1337;
    int strength = 100;
    int defense  = 200;
}
```

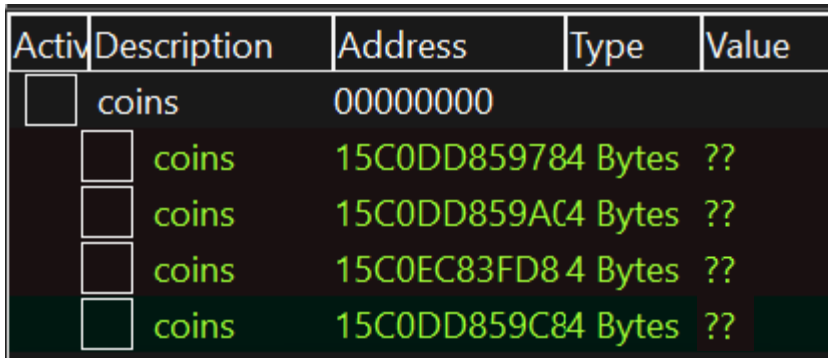
Data structures

Description	Address	Type	Value
health	15C0DD85978	4 Bytes	1337




```
struct Player
{
    int health   = 1337;
    int strength = 100;
    int defense  = 200;
}
```

Adresses mémoire persistantes.



On recharge le jeu
On perd tout!

Adresses mémoire persistantes.

Solution: chercher des adresses *potentiellement* statiques

recherche toutes les instructions pointant vers cette adresse

```
coins                    2216AF30F68     4 Bytes  58
```

Generate pointermap

Recursive scan

```
op [(Address - 1) + 01]
op [(Address - 2) + 02]
op [(Address - 3) + 03]
...
```

```
7FF7BB2E2724 - 48 03 41 08  - add rax,[rcx+08]      RCX=000002216AF30F60
7FF7BB165E68 - 49 89 44 24 08  - mov [r12+08],rax   R12=000002216AF30F60
7FF7BB1B9D40 - 49 8B 55 08  - mov rdx,[r13+08]      R13=000002216AF30F60
```

Adresses mémoire persistantes.

Problème: trop de résultats!

4 Bytes | Pointer paths 230860

Base Address	Offset 0	Points to:
"godot.windows.opt.tools.64.exe"+07212940	10	-
"godot.windows.opt.tools.64.exe"+071DE070	60	-
"godot.windows.opt.tools.64.exe"+07212940	10	-
"godot.windows.opt.tools.64.exe"+071DE070	60	-

Adresses mémoire persistantes.

solution: rescan, et comparer les résultats

Filename	Address
pointermap_coins5.scandata	248C053F6E8
pointermap_coins3.scandata	1D847EDC898
<Select a file>	

4 Bytes | Pointer paths 1

Base Address	Offset 0	Points to:
"godot.windows.opt.tools.64.exe"+0717F820	3B8	2216AF30F68 = 206

coins	2216AF30F68	4 Bytes	74
pointerscan result	P->2216AF30F68	4 Bytes	74

Et si on ne cherche pas une valeur?

Comment chercher une condition?

```
def player_move():
    if collision("coin"):
        coins += 1
    if collision("door"):
        if has_key:
            open_door()
    if button("down"):
        crouch()
```

Et si on ne cherche pas une valeur?

Comment chercher une condition:
code filter

Memory View | Tools | Code Filter

Addresses executed since last filter operation:0

Has been executed

Has not been executed

Start | Stop

Load address list

From Trace

From Disassembler

From File

Address List (46093)

Address	Executed
Tutorial-i386.exe.text+1BB5	No
Tutorial-i386.exe.text+1BBA	No
Tutorial-i386.exe.text+1BBF	No
Tutorial-i386.exe.text+1BC8	No

```
def player_move():
    if collision("coin"):
        coins += 1
    if collision("door"):
        if has_key:
            open_door()
    if button("down"):
        crouch()
```

Et si on ne cherche pas une valeur?

Comment chercher une condition:
code filter

Addresses executed since last filter operation:1791

Has been executed

Has not been executed

Start | Stop

Address List (44302)

Address	Executed
Tutorial-i386.exe.text+1BB5	No
Tutorial-i386.exe.text+1BBA	No
Tutorial-i386.exe.text+1BBF	No
Tutorial-i386.exe.text+1BC8	Yes

```
def player_move():
    if collision("coin"):
        coins += 1
    if collision("door"):
        if has_key:
            open_door()
    if button("down"):
        crouch()
```

Et si on ne cherche pas une valeur?

Comment chercher une condition:
code filter

Addresses executed since last filter operation: 595

Has been executed

Has not been executed

Start | Stop

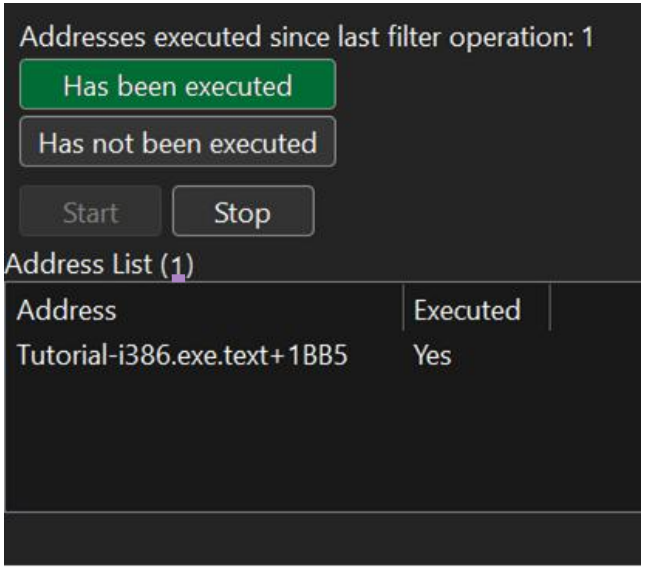
Address List (595)

Address	Executed
Tutorial-i386.exe.text+1BB5	No
Tutorial-i386.exe.text+1BBA	No
Tutorial-i386.exe.text+1BBF	No
Tutorial-i386.exe.text+1BC8	Yes

```
def player_move():
    if collision("coin"):
        coins += 1
    if collision("door"):
        if has_key:
            open_door()
    if button("down"):
        crouch()
```

Et si on ne cherche pas une valeur?

Comment chercher une condition:
code filter




```
def player_move():
    if collision("coin"):
        coins += 1
    if collision("door"):
        if has_key:
            open_door()
    if button("down"):
        crouch()
```

Instruction patching

de l'ASM, oskour!

```
74 02          je    Tutorial-i386.exe.text+26687
EB 49          jmp   Tutorial-i386.exe.text+266D0
A1 B0666500    mov   eax,[Tutorial-i386.exe+2566B0]
3B 45 E8       cmp   eax,[ebp-18]
74 02          je    Tutorial-i386.exe.text+26693
EB 1F          jmp   Tutorial-i386.exe.text+266B2
C7 45 E8 000...mov   [ebp-18],00000000
6A 00          push  00
```

Instruction patching

ASM primer

je if ==	**jg** if >	**add** +=
jne if !==	**jl** if <	**sub** -=

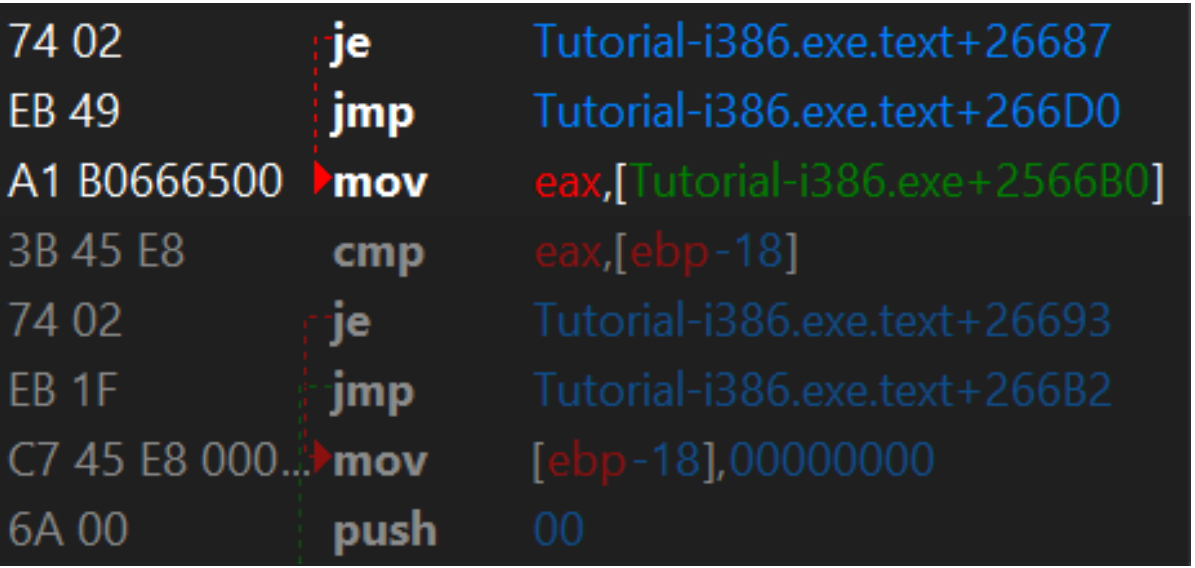
mov x=y

nop do nothing

(padding)

Instruction patching

Remplacer la condition *has_key*




```
def player_move():
    if collision("coin"):
        coins += 1
    if collision("door"):
        if has_key:
            open_door()
    if button("down"):
        crouch()
```

Instruction patching

Remplacer la condition *has_key*

```
74 02          je      Tutorial-i386.exe.text+26687
EB 49          jmp     Tutorial-i386.exe.text+266D0
A1 B0666500    mov     eax,[Tutorial-i386.exe+2566B0]
3B 45 E8       cmp     eax,[ebp-18]
74 02          je      Tutorial-i386.exe.text+26693
EB 1F          jmp     Tutorial-i386.exe.text+266B2
C7 45 E8 000...mov     [ebp-18],00000000
6A 00          push    00
```

```
def player_move():
    if collision("coin"):
        coins += 1
    if collision("door"):
        if has_key:
            open_door()
    if button("down"):
        crouch()
```

Instruction patching

Remplacer la condition *has_key*


```
def player_move():
    if collision("coin"):
        coins += 1
    if collision("door"):
        if not has_key:
            open_door()
    if button("down"):
        crouch()
```

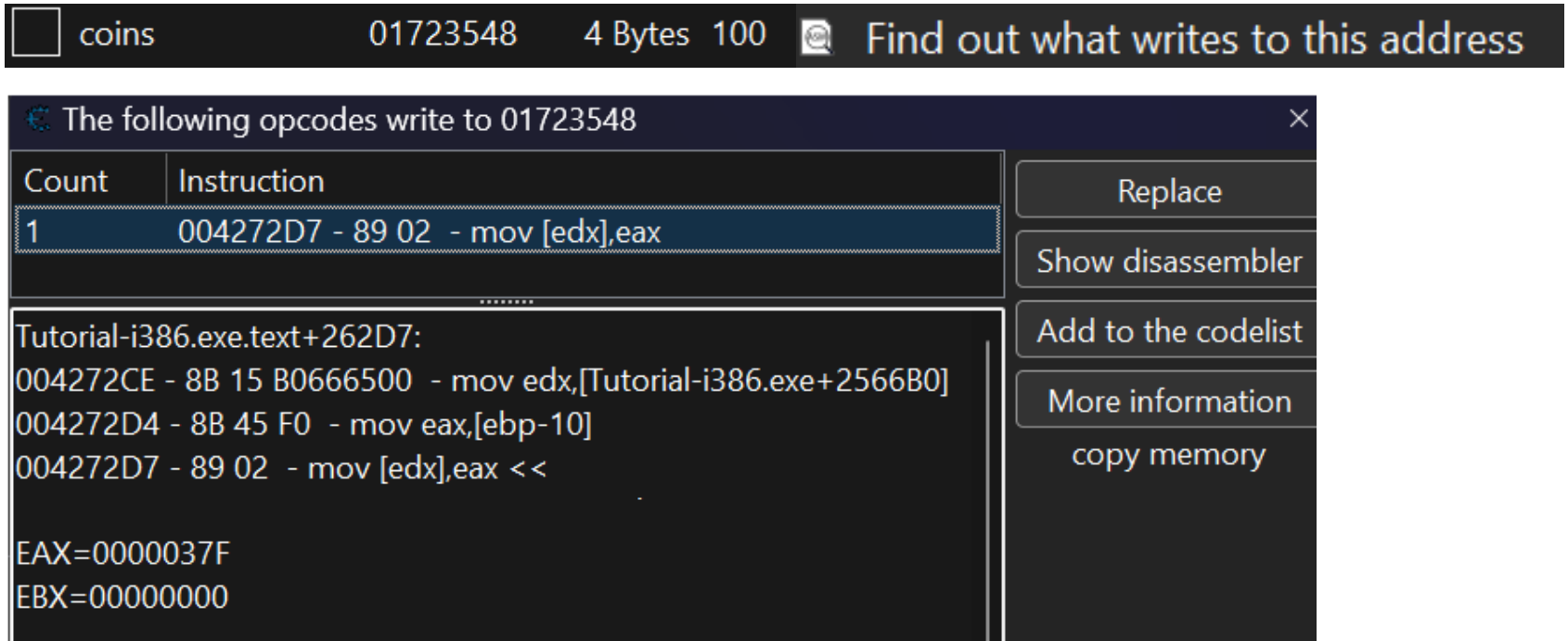
Instruction patching

Remplacer la condition *has_key*

```
74 02          je    Tutorial-i386.exe.text+26687
EB 49          jmp   Tutorial-i386.exe.text+266D0
A1 B0666500    mov   eax,[Tutorial-i386.exe+2566B0]
3B 45 E8       cmp   eax,[ebp-18]
75 02          jne   Tutorial-i386.exe.text+26693
EB 1F          jmp   Tutorial-i386.exe.text+266B2
C7 45 E8 000...mov   [ebp-18],00000000
6A 00          push  00
```

Instruction patching: problèmes

Et si on partait d'adresses connues?



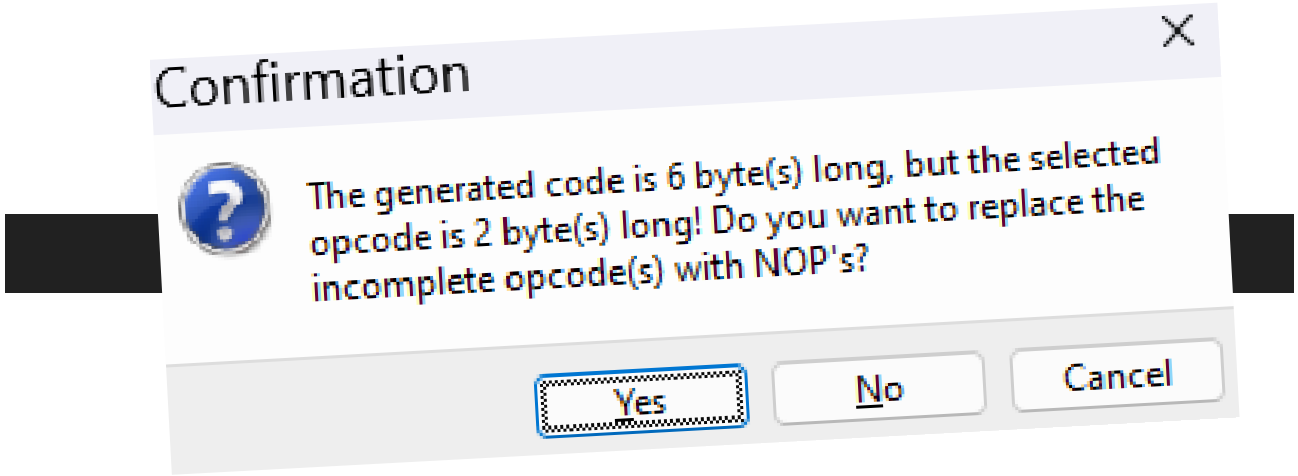
Instruction patching: problèmes

Instructions moins évidentes

```
mov     [edx],eax
```

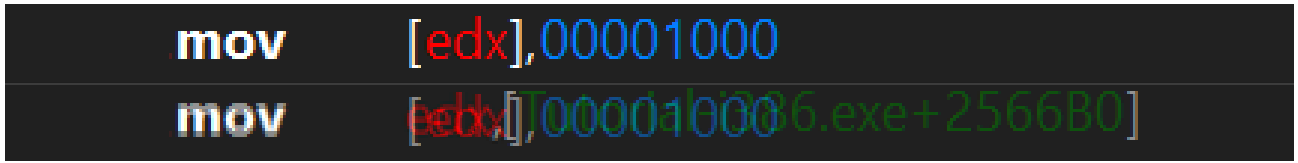
Instruction patching: problèmes

Clash de tailles d'instructions



Instruction patching: problèmes

Clash de tailles d'instructions



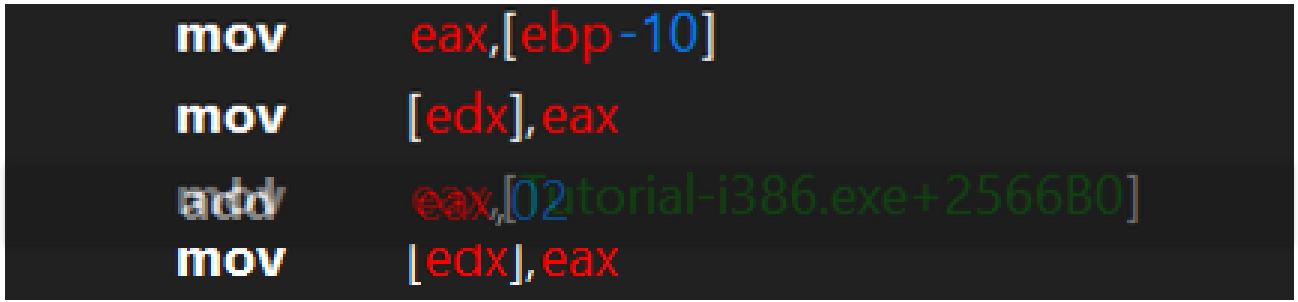
Instruction patching: problèmes

Ou rajouter notre code?

```
add     eax,02
mov     [edx],eax
```

Instruction patching: problèmes

Ou rajouter notre code?



Autoassemble

Solution: **auto assembler!**

- Auto-alloue de la mémoire
- Créer des methodes complexes
- Persistant et désactivable

```
[ENABLE]
alloc(newmem,2048)
label(return)
alloc(multiplier, 2)
registerSymbol(multiplier)



multiplier:
    dd (int)5

newmem:
mov eax, [edx]
add eax, [multiplier]
mov [edx],eax
mov eax, [Tutorial-i386.exe+2566B0]
jmp return

"Tutorial-i386.exe"+272D7:
jmp newmem
nop 2
return:

[DISABLE]
dealloc(multiplier)
unregisterSymbol(multiplier)

dealloc(newmem)
"Tutorial-i386.exe"+272D7:
db 89 02 A1 B0 66 65 00
```

Autoassemble

Solution: **auto assembler!**

- **Auto-alloue de la mémoire**

```
alloc(newmem,2048)
label(return)

newmem:
// your code here

jmp return

"Tutorial-i386.exe"+272D7: //original
jmp newmem                 address
nop 2
return:
```

Autoassemble

Solution: **auto assembler!**

- Auto-alloue de la mémoire

  gères labels, variables…


```
alloc(newmem,2048)
label(return)
alloc(multiplier, 2)
registerSymbol(multiplier)



multiplier:
    dd (int)5

newmem:
mov eax, [edx]
add eax, [multiplier]


jmp return

"Tutorial-i386.exe"+272D7:
jmp newmem
nop 2
return:
```

Autoassemble

Solution: **auto assembler!**

- Auto-alloue de la mémoire
- **Créer des methodes complexes**

```
alloc(newmem,2048)
label(return)
alloc(multiplier, 2)
registerSymbol(multiplier)



multiplier:
    dd (int)5

newmem:
mov eax, [edx] //coins += multiplier
add eax, [multiplier]
mov [edx],eax
mov eax, [Tutorial-i386.exe+2566B0]
jmp return

"Tutorial-i386.exe"+272D7:
jmp newmem
nop 2
return:
```

Autoassemble

Solution: **auto assembler!**

- Auto-alloue de la mémoire
- Créer des methodes complexes
- **Persistant et désactivable**

```
[ENABLE]
alloc(newmem,2048)
label(return)
alloc(multiplier, 2)
registerSymbol(multiplier)




multiplier:
    dd (int)5

newmem:
mov eax, [edx]
add eax, [multiplier]
mov [edx],eax
mov eax, [Tutorial-i386.exe+2566B0]
jmp return

"Tutorial-i386.exe"+272D7:
jmp newmem
nop 2
return:

[DISABLE]
dealloc(multiplier)
unregisterSymbol(multiplier)

dealloc(newmem)
"Tutorial-i386.exe"+272D7:
db 89 02 A1 B0 66 65 00
```

Autoassemble

Solution: **auto assembler!**

- Auto-alloue de la mémoire
- Créer des methodes complexes
- **Persistant et désactivable**

```
[ENABLE]
alloc(newmem,2048)
label(return)
alloc(multiplier, 2)
registerSymbol(multiplier)




multiplier:
    dd (int)5

newmem:
mov eax, [edx]
add eax, [multiplier]
mov [edx],eax
mov eax, [Tutorial-i386.exe+2566B0]
jmp return

"Tutorial-i386.exe"+272D7:
jmp newmem
nop 2
return:

[DISABLE]
dealloc(multiplier)
unregisterSymbol(multiplier)

dealloc(newmem)
"Tutorial-i386.exe"+272D7:
db 89 02 A1 B0 66 65 00
```

Autoassemble

Solution: **auto assembler!**

- Auto-alloue de la mémoire
- Créer des methodes complexes
- Persistant et désactivable

Resiste aux changements de binaires avec scans AOB

```
[ENABLE]
alloc(newmem,2048)
label(return)
alloc(multiplier, 2)
registerSymbol(multiplier)
registerSymbol(INJECT)
aobscanmodule(INJECT,Tutorial-i386.exe,
                89 02 A1 B0 66 65 00)

multiplier:
    dd (int)5

newmem:
mov eax, [edx]
add eax, [multiplier]
mov [edx],eax
mov eax, [Tutorial-i386.exe+2566B0]
jmp return

INJECT:
jmp newmem
nop 2
return:

[DISABLE]
dealloc(multiplier)
unregisterSymbol(multiplier)

dealloc(newmem)
INJECT:
db 89 02 A1 B0 66 65 00
unregistersymbol(INJECT)
```

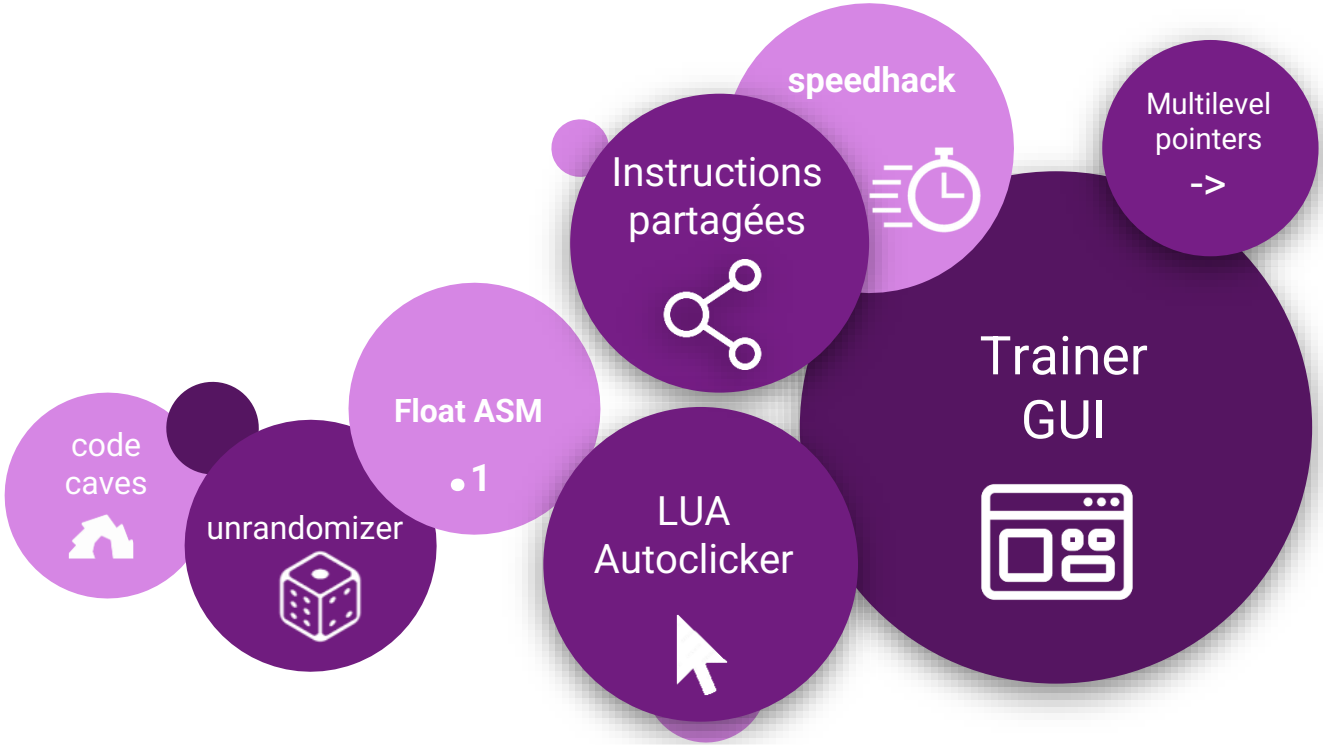
Autoassemble

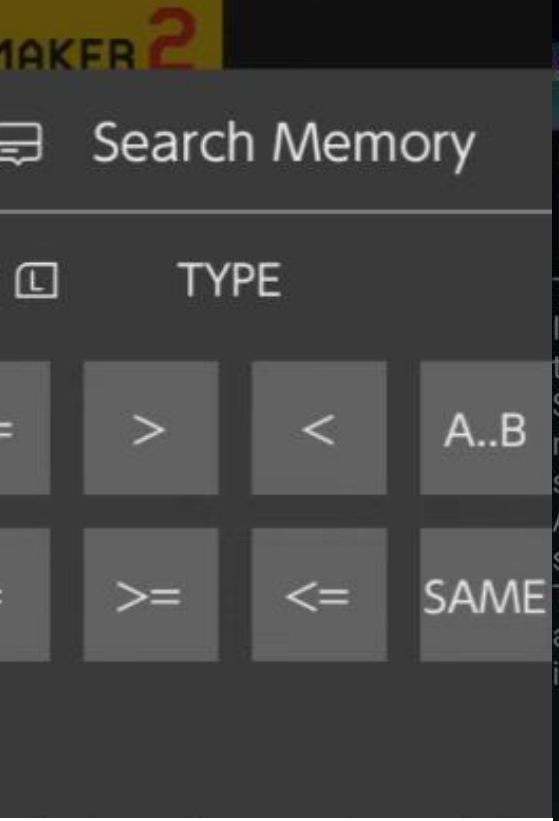
Solution: **auto assembler!**

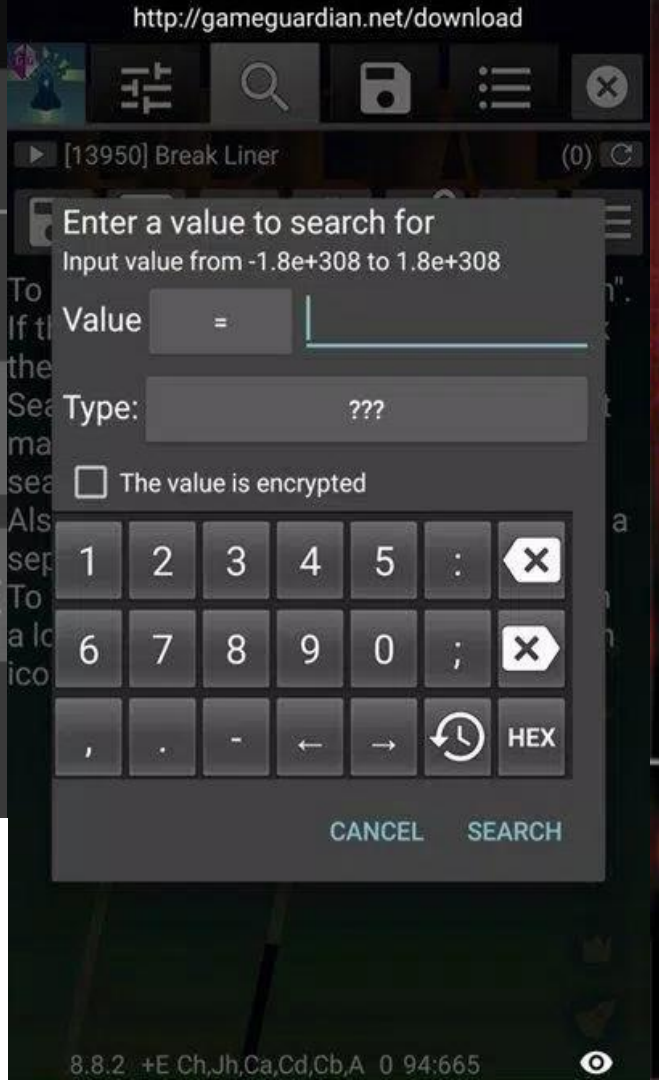
- Auto-alloue de la mémoire
- Créer des methodes complexes
- Persistant et désactivable

```
[ENABLE]
alloc(newmem,2048)
label(return)
alloc(multiplier, 2)
registerSymbol(multiplier)
registerSymbol(INJECT)
aobscanmodule(INJECT,Tutorial-i386.exe,
                89 02 A1 B0 66 65 00)

multiplier:
    dd (int)5

newmem:
mov eax, [edx]
add eax, [multiplier]
mov [edx],eax
mov eax, [Tutorial-i386.exe+2566B0]
jmp return

INJECT:
jmp newmem
nop 2
return:

[DISABLE]
dealloc(multiplier)
unregisterSymbol(multiplier)

dealloc(newmem)
INJECT:
db 89 02 A1 B0 66 65 00
unregistersymbol(INJECT)
```

Next steps


speedhack
Instructions partagées
Multilevel pointers ->
Trainer GUI
code caves
unrandomizer
Float ASM
• 1
LUA Autoclicker



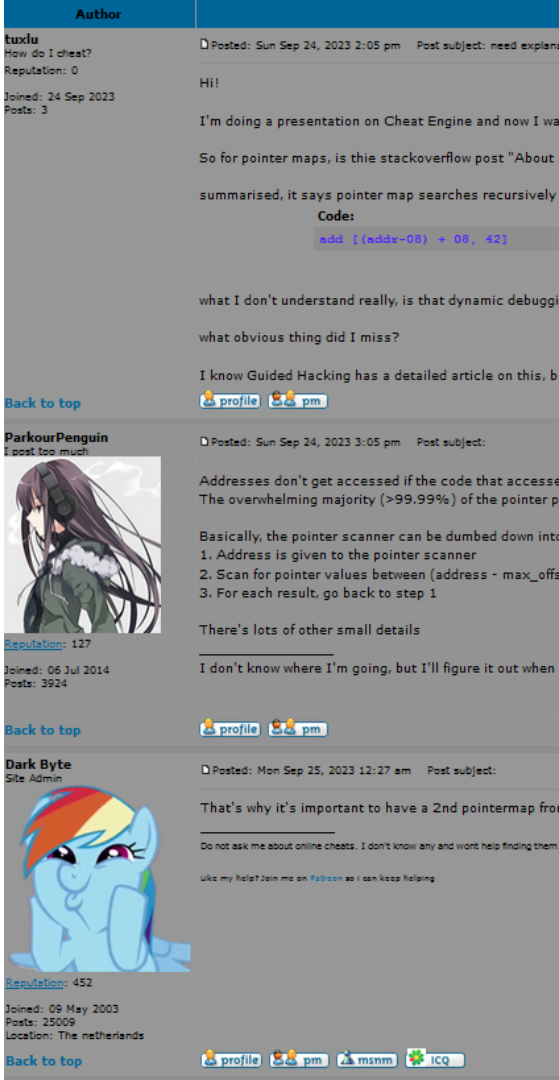


Equivalents sur toutes platformes

Plus compliqués à setup

Resources

- Forums Cheat Engine



Author

tuxlu
How do I cheat?
Reputation: 0
Joined: 24 Sep 2023
Posts: 3

Posted: Sun Sep 24, 2023 2:05 pm Post subject: need explana

Hi!

I'm doing a presentation on Cheat Engine and now I wa

So for pointer maps, is thie stackoverflow post "About

summarised, it says pointer map searches recursively

Code:

```
add [(addr-08) + 08, 42]
```

what I don't understand really, is that dynamic debuggi

what obvious thing did I miss?

I know Guided Hacking has a detailed article on this, b

Back to top profile pm

ParkourPenguin
I post too much

Reputation: 127
Joined: 06 Jul 2014
Posts: 3924

Posted: Sun Sep 24, 2023 3:05 pm Post subject:

Addresses don't get accessed if the code that accesse
The overwhelming majority (>99.99%) of the pointer p

Basically, the pointer scanner can be dumbed down into
1. Address is given to the pointer scanner
2. Scan for pointer values between (address - max_offs
3. For each result, go back to step 1

There's lots of other small details

I don't know where I'm going, but I'll figure it out when

Back to top profile pm

Dark Byte
Site Admin

Reputation: 452
Joined: 09 May 2003
Posts: 25009
Location: The netherlands

Posted: Mon Sep 25, 2023 12:27 am Post subject:

That's why it's important to have a 2nd pointermap fro

Do not ask me about online cheats. I don't know any and wont help finding them

Like my help? Join me on Patreon so I can keep helping

Back to top profile pm msnm ICQ

Resources

- Forums Cheat Engine

- **Vidéos Youtube**
  Par Stephen Chapman
  et Guided Hacking

Resources

- Forums Cheat Engine

- Vidéos Youtube
  Par Stephen Chapman
  et Guided Hacking

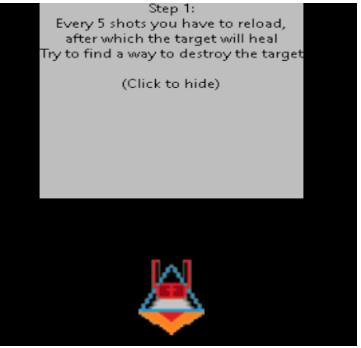
- **challenges jeux vidéos!**

3

Créer un jeu pour les hackers

Shall we play a game?

Jeux CtF existants



Cheat Engine tuto Built-in

Un peu basique…

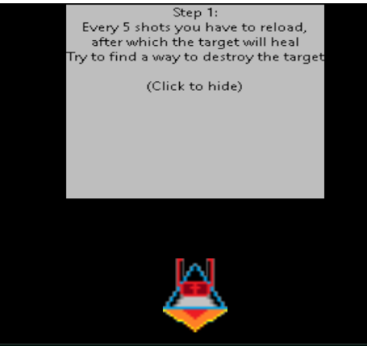
Jeux CtF existants

Cheat Engine
tuto Built-in

Pwn Adventure

Un peu trop
compliqué à setup!

Jeux CtF existants



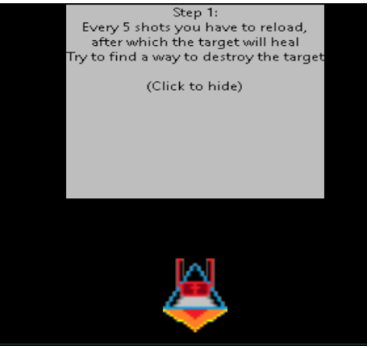
Cheat Engine
tuto Built-in

Pwn Adventure

Google CTF
Hackceler8

Pas que du
game hacking

Jeux CtF existants



Cheat Engine
tuto Built-in

Pwn Adventure

Google CTF
Hackceler8

Rootme's HackerMan

Sorti 2 jours
avant mon jeu ^^'

FLAG QUEST

HACKDAY

devoteam
Cyber Trust

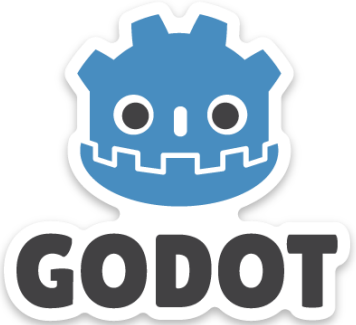
trailer music: Robyn - touch (edited)

Godot Engine

Moteur libre et open source

Léger et iteration rapide

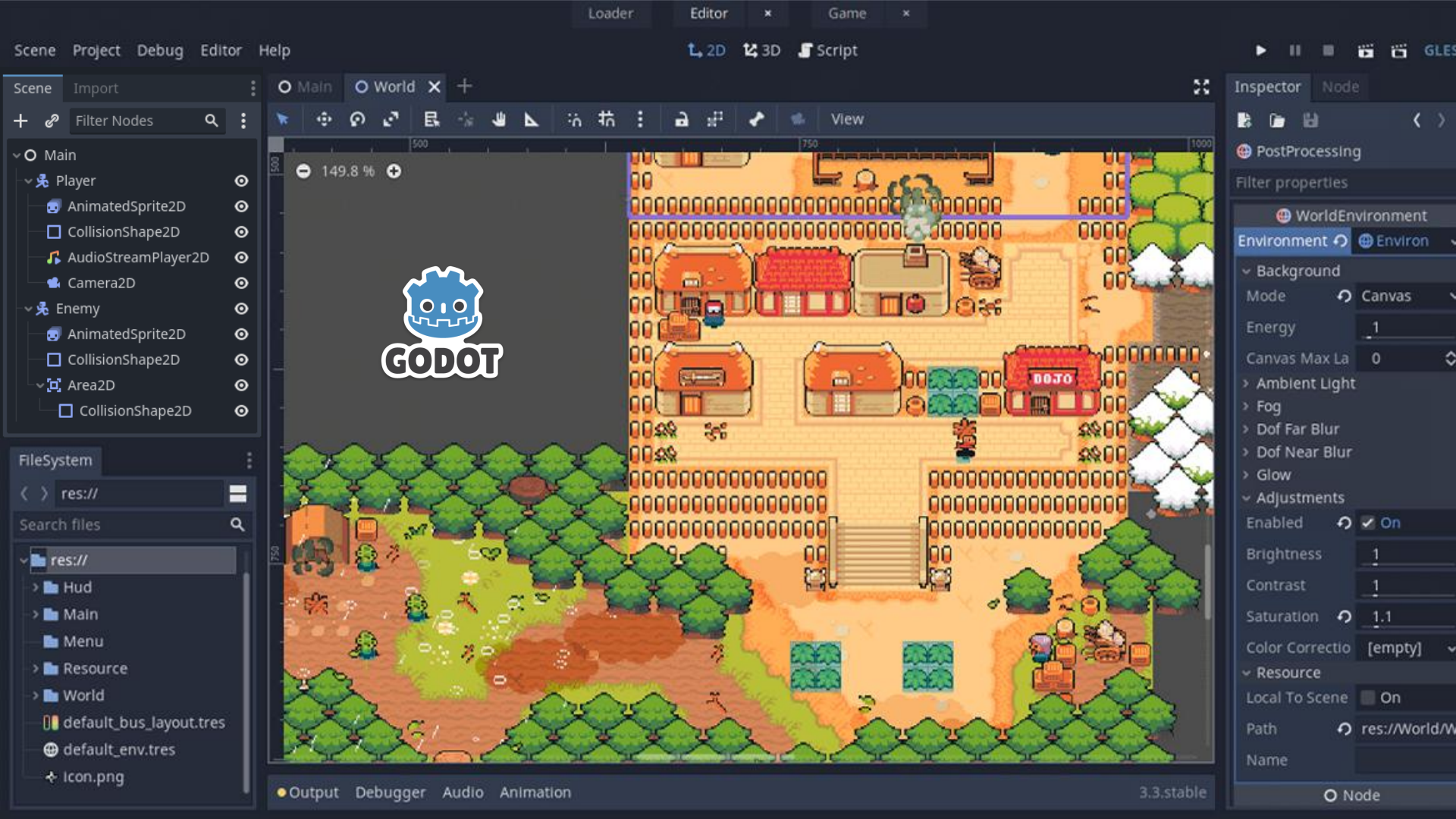
Communauté grandissante

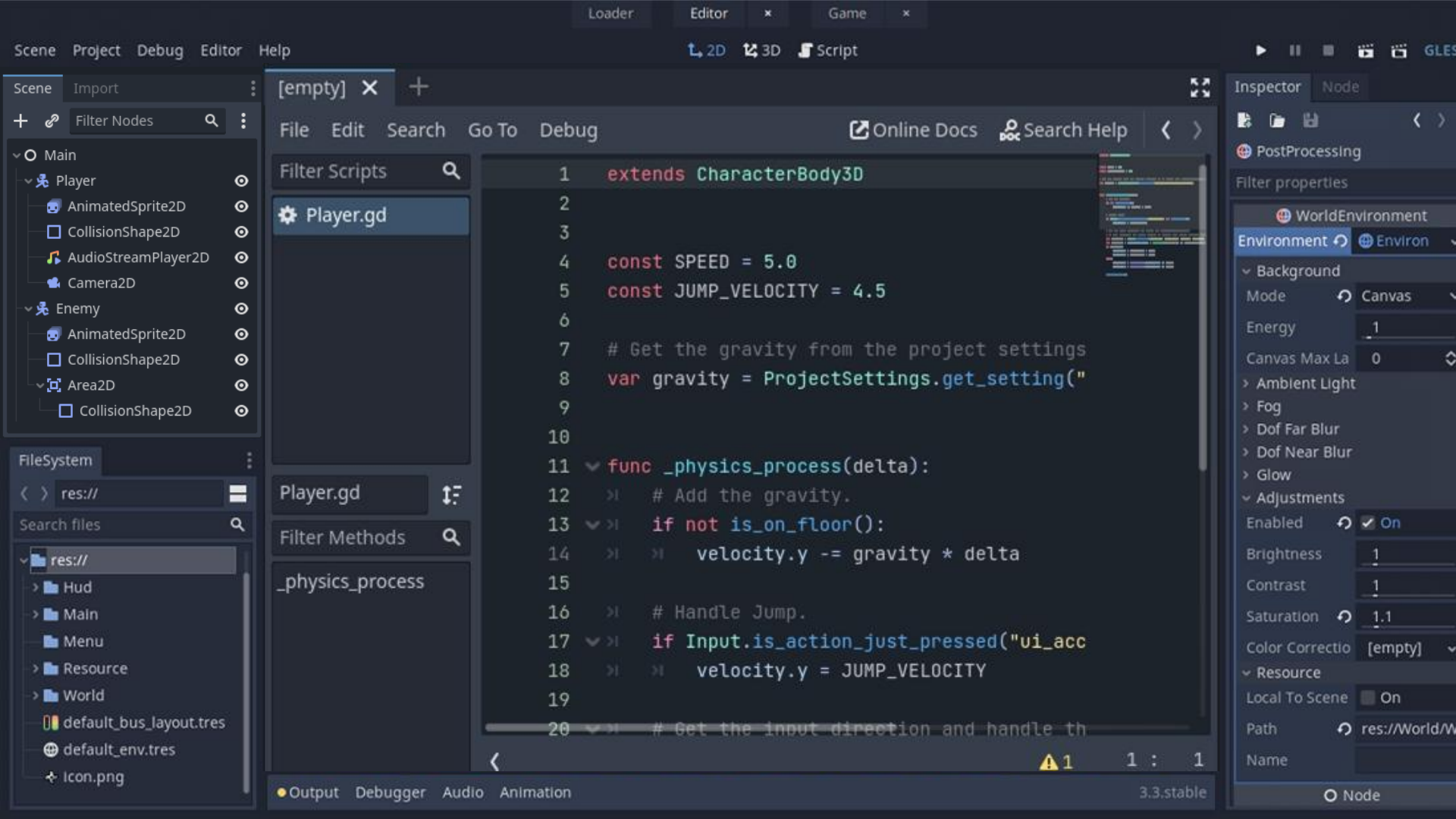


Pas encore là pour la grosse 3D

Performances améliorables

Manque de features avancées


Loader
Editor
Game
Scene
Project
Debug
Editor
Help
2D
3D
Script
GLES
Scene
Import
Filter Nodes
Main
Player
AnimatedSprite2D
CollisionShape2D
AudioStreamPlayer2D
Camera2D
Enemy
AnimatedSprite2D
CollisionShape2D
Area2D
CollisionShape2D
FileSystem
res://
Search files
res://
Hud
Main
Menu
Resource
World
default_bus_layout.tres
default_env.tres
Icon.png
Main
World
View
149.8 %
GODOT
DOJO
Output
Debugger
Audio
Animation
3.3.stable
Inspector
Node
PostProcessing
Filter properties
WorldEnvironment
Environment
Environ
Background
Mode
Canvas
Energy
1
Canvas Max La
0
Ambient Light
Fog
Dof Far Blur
Dof Near Blur
Glow
Adjustments
Enabled
On
Brightness
1
Contrast
1
Saturation
1.1
Color Correctio
[empty]
Resource
Local To Scene
On
Path
res://World/W
Name
Node


Loader
Editor
Game
Scene Project Debug Editor Help
2D 3D Script
Scene Import
Filter Nodes
Main
Player
AnimatedSprite2D
CollisionShape2D
AudioStreamPlayer2D
Camera2D
Enemy
AnimatedSprite2D
CollisionShape2D
Area2D
CollisionShape2D
FileSystem
res://
Search files
res://
Hud
Main
Menu
Resource
World
default_bus_layout.tres
default_env.tres
Icon.png
[empty]
File Edit Search Go To Debug
Online Docs
Search Help
Filter Scripts
Player.gd
Player.gd
Filter Methods
_physics_process
1 extends CharacterBody3D
2
3
4 const SPEED = 5.0
5 const JUMP_VELOCITY = 4.5
6
7 # Get the gravity from the project settings
8 var gravity = ProjectSettings.get_setting("
9
10
11 func _physics_process(delta):
12 # Add the gravity.
13 if not is_on_floor():
14 velocity.y -= gravity * delta
15
16 # Handle Jump.
17 if Input.is_action_just_pressed("ui_acc
18 velocity.y = JUMP_VELOCITY
19
20 # Get the input direction and handle th
1
1 : 1
Output Debugger Audio Animation
3.3.stable
Inspector Node
PostProcessing
Filter properties
WorldEnvironment
Environment
Environ
Background
Mode
Canvas
Energy
1
Canvas Max La
0
Ambient Light
Fog
Dof Far Blur
Dof Near Blur
Glow
Adjustments
Enabled
On
Brightness
1
Contrast
1
Saturation
1.1
Color Correctio
[empty]
Resource
Local To Scene
On
Path
res://World/W
Name
Node

4

Protéger votre jeu des hackers

No fun allowed

Protéger votre jeu

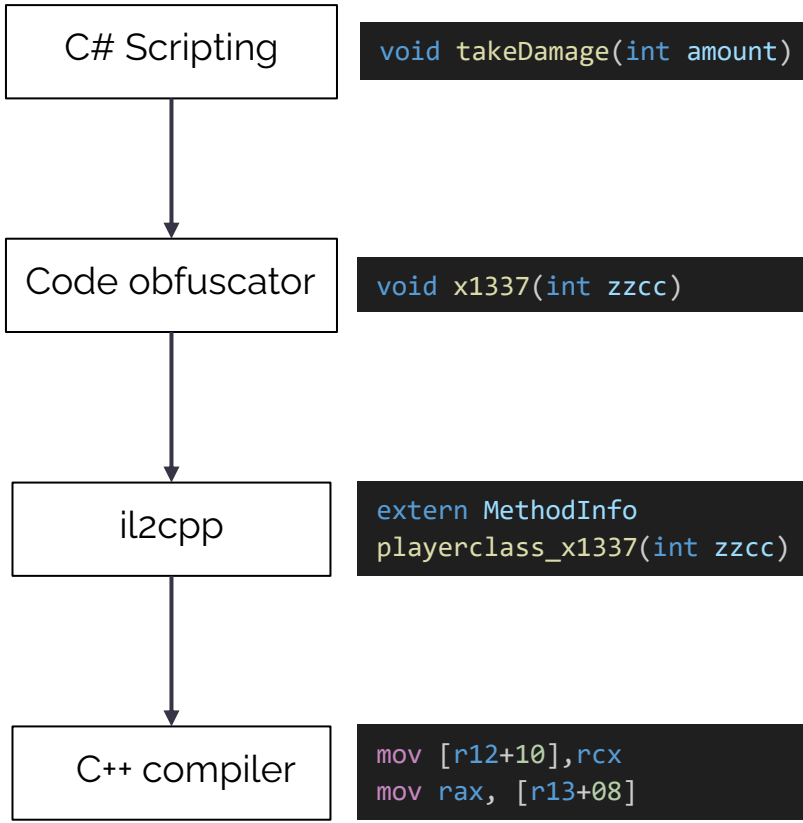
- Obfusquer les valeurs en mémoire

```
struct AntiCheatInt
{
    int projected;
    int r = rand();

    public int Value {
        get => (projected + r) / 3;
        set => projected = (value * 3) - r;
    }
}
```

Protéger votre jeu

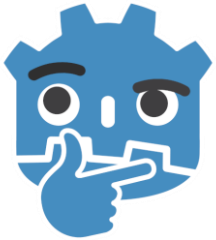
- Obfusquer les valeurs en mémoire
- **Obfusquer les binaires**

unity Code pipeline



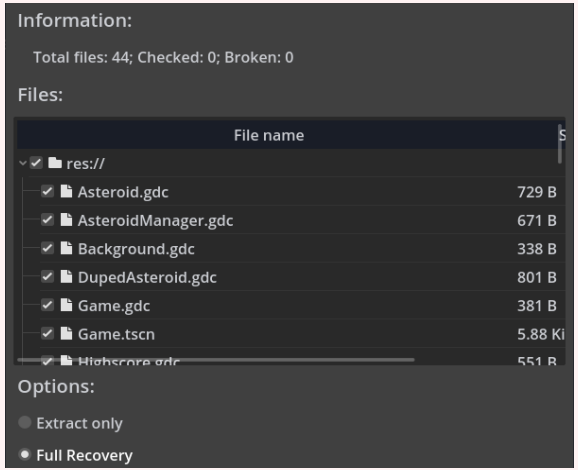
Protéger votre jeu: Godot

Langage interprété et
format open source



Protéger votre jeu: Godot

Langage interprété et format open source

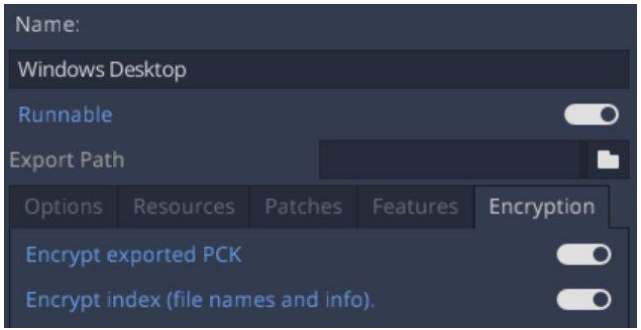


Décompilable entièrement avec GdsDecomp

Protéger votre jeu: Godot

Langage interprété et format open source

Le jeu peut être chiffré avec une clé AES

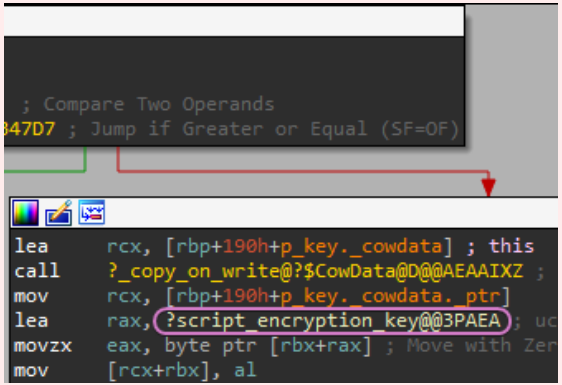


Décompilable entièrement avec GdsDecomp

Protéger votre jeu: Godot

Langage interprété et format open source

Le jeu peut être chiffré avec une clé AES



Décompilable entièrement avec GdsDecomp

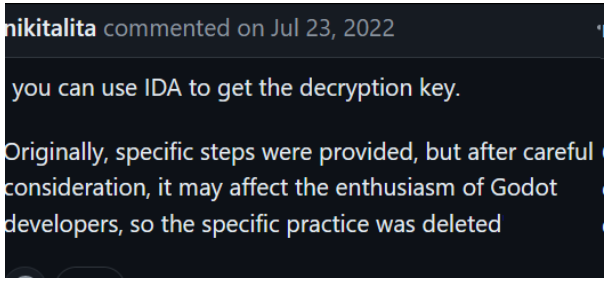
La clé est extractable dans le binaire

Protéger votre jeu: Godot

Langage interprété et format open source

Le jeu peut être chiffré avec une clé AES

Le dev de GdsDecomp ne donnera pas de doc sur comment l'extraire



Décompilable entièrement avec GdsDecomp

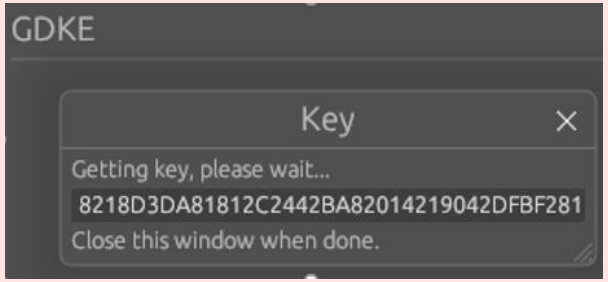
La clé est extractable dans le binaire

Protéger votre jeu: Godot

Langage interprété et format open source

Le jeu peut être chiffré avec une clé AES

Le dev de GdsDecomp ne donnera pas de doc sur comment l'extraire



Décompilable entièrement avec GdsDecomp

La clé est extractable dans le binaire

Quelqu'un d'autre en a fait un tool: gdke

Protéger votre jeu: Godot

Langage interprété et format open source

Le jeu peut être chiffré avec une clé AES

Le dev de GdsDecomp ne donnera pas de doc sur comment l'extraire

On peut modifier quelques lignes du moteur pour fool le tool

```
Vector<uint8_t> p_key = raw_key.reverse();

std::transform(p_key.begin(), p_key.end(),
p_xor_key.begin(), p_key.begin(),
std::bit_xor<uint8_t>());
```

Décompilable entièrement avec GdsDecomp

La clé est extractable dans le binaire

Quelqu'un d'autre en a fait un tool: gdke

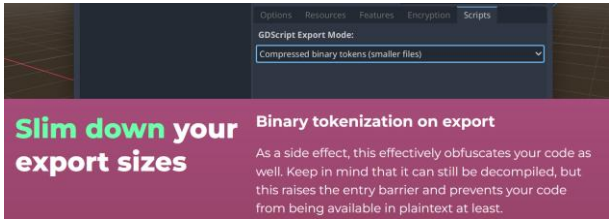
Protéger votre jeu: Godot

Langage interprété et format open source

Le jeu peut être chiffré avec une clé AES

Le dev de GdsDecomp ne donnera pas de doc sur comment l'extraire

On peut modifier quelques lignes du moteur pour fool le tool



Décompilable entièrement avec GdsDecomp

La clé est extractable dans le binaire

Quelqu'un d'autre en a fait un tool: gdke

Toujours trouvable sur Ghidra pour un reverser motivé

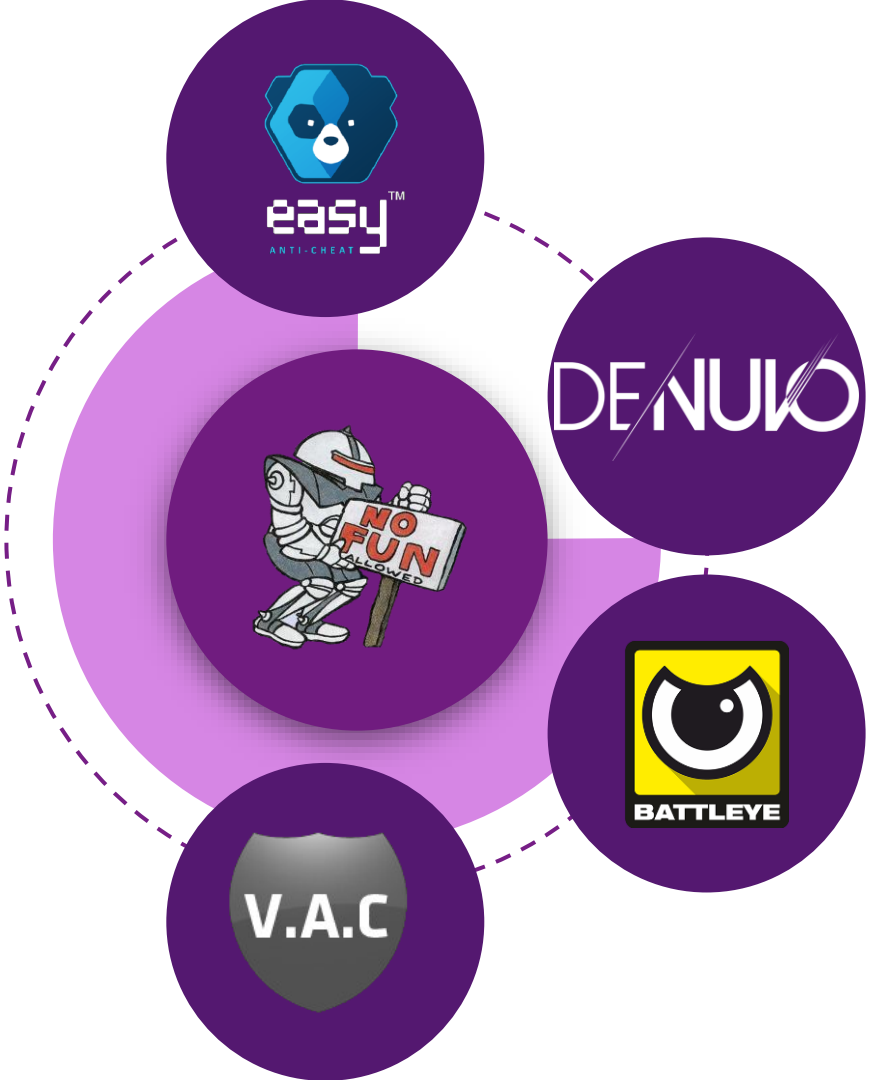
Protéger votre jeu

- Obfusquer les valeurs en mémoire
- Obfusquer les binaires
- **Chiffrer les binaires et sauvegardes**

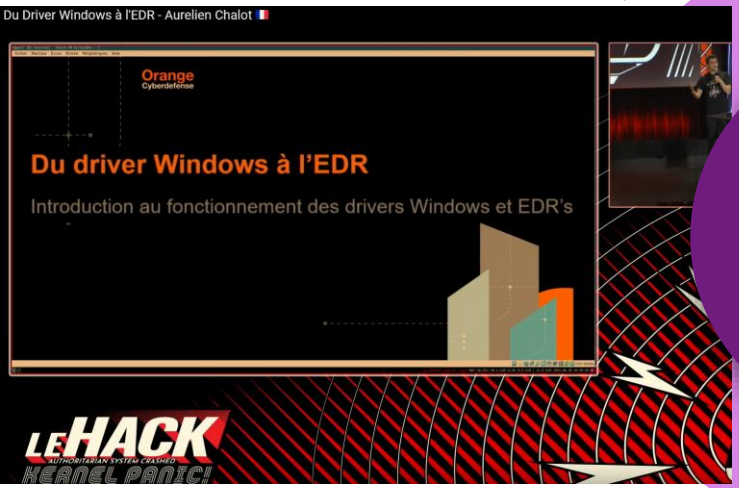
Protéger votre jeu

- Obfusquer les valeurs en mémoire
- Obfusquer les binaires
- Chiffrer les binaires et sauvegardes
- **Ne pas trust le client:**
  tout verifier server side

Logiciels
Anti cheat



Logiciels
Anti cheat



Analyse tous les process

Détecte hook fonctions

EDR

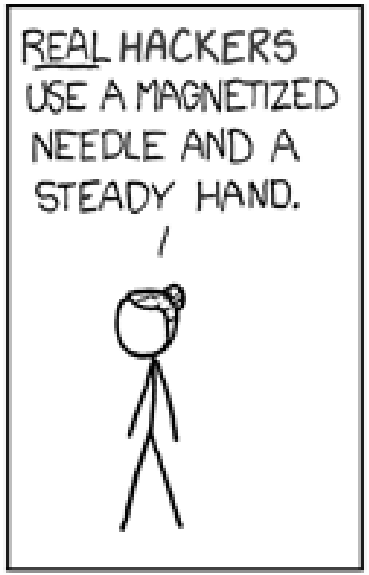
En gros

Check mémoire

driver Kernel mode

Advanced cheating:
Sans toucher le jeu.

- **Hacks hardware,**
  plus difficile à détecter.



Advanced cheating:
Sans toucher le jeu.

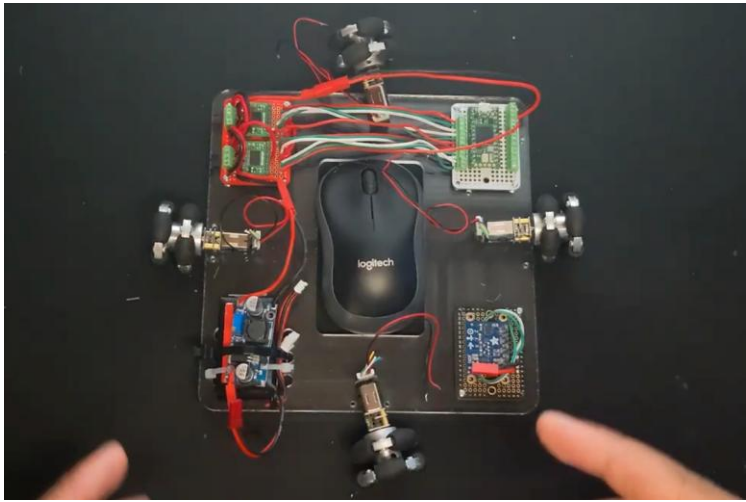
- Hacks hardware,
  plus difficile à détecter.

- **Tools de lecture d'écran (aimbot)**

Advanced cheating:
Sans toucher le jeu.

- Hacks hardware,
  plus difficile à détecter.

- **Tools de lecture d'écran (aimbot)**



Advanced cheating:
Sans toucher le jeu.

- Hacks hardware,
  plus difficile à détecter.

- Tools de lecture d'écran (aimbot)

- **désynchronisation via latence en ligne**

Advanced cheating:
Sans toucher le jeu.

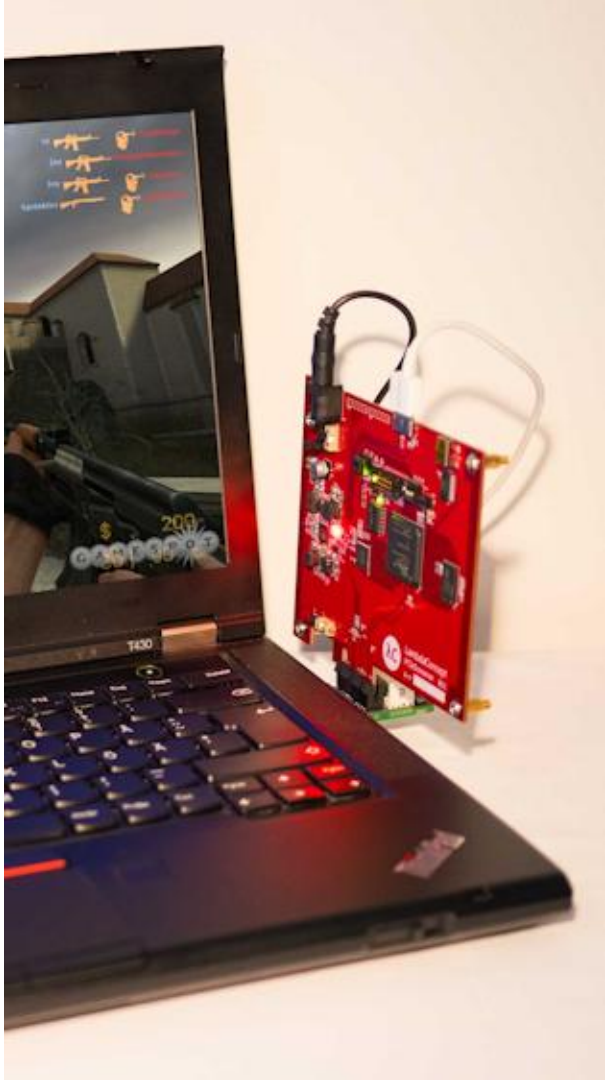
- Hacks hardware,
  plus difficile à détecter.

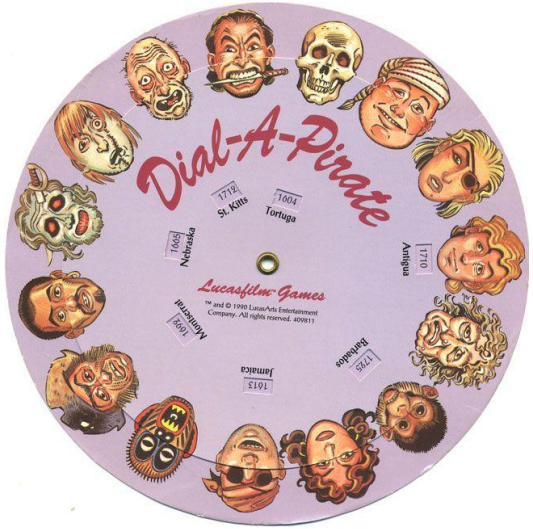
- Tools de lecture d'écran (aimbot)

- **désynchronisation via latence en ligne**

Advanced cheating:
Sans toucher le jeu.

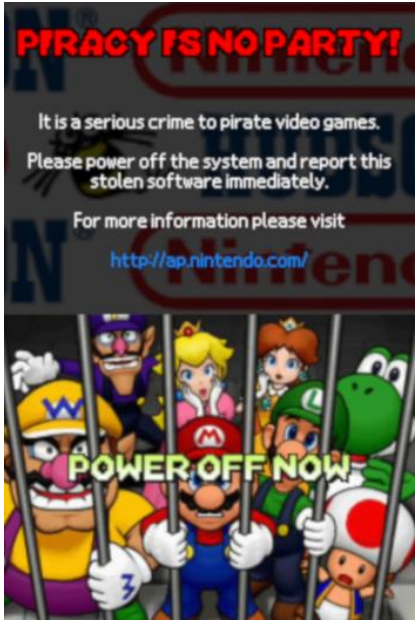
- Hacks hardware,
  plus difficile à détecter.
- Tools de lecture d'écran (aimbot)
- désynchronisation via latence en ligne
- **Direct Memory Access via PCIE**





tuxlu.fr/talk_vghacking

tuxlu.fr/talk_vghacking



thank you.