

# 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



8% ALWAYS



9% OFTEN



18% SOMETIMES



12% RARELY



57% NEVER

# 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

Bypass Anti-cheat  
Code Filter  
Détection de valeurs  
 **Scan mémoire**  
Debugging  
Cheat Engine

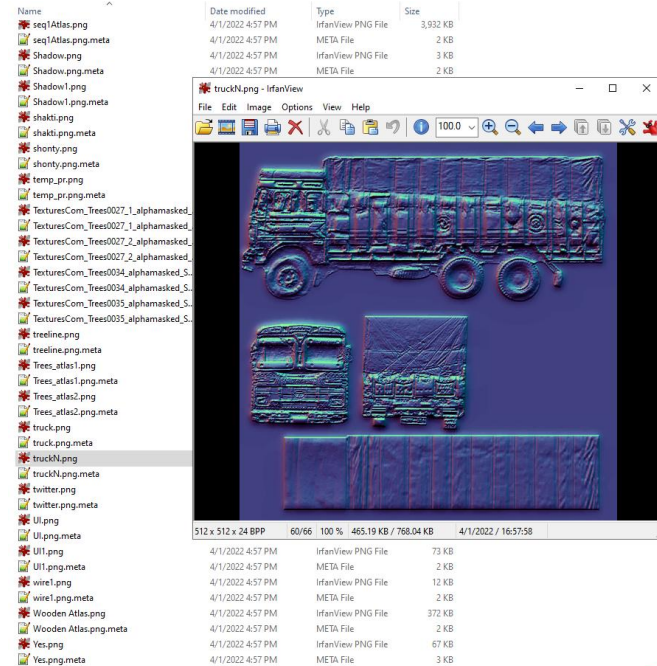
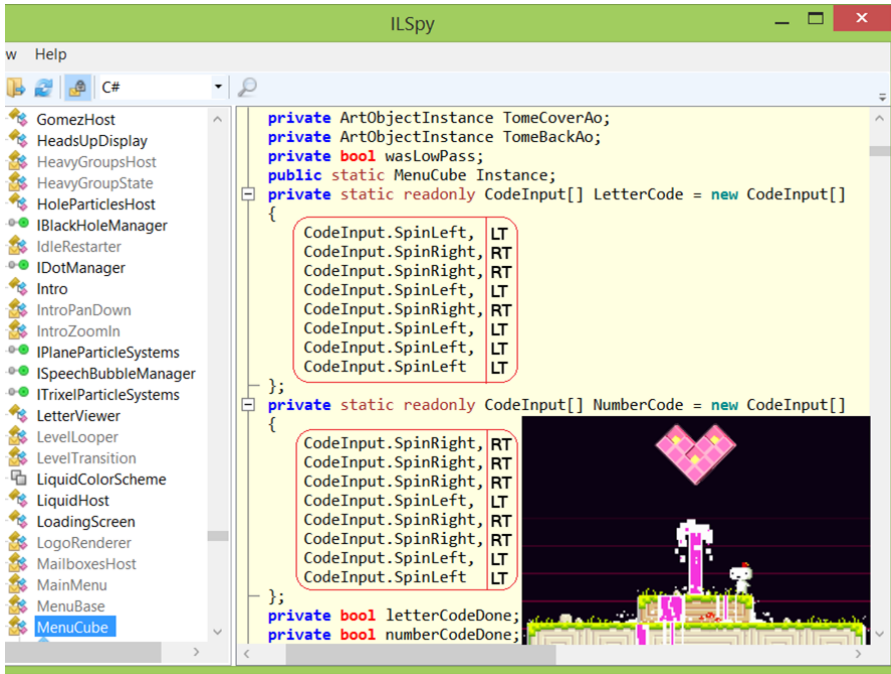
Vérifications serveur  
Pwn multijoueur  
 **Interception réseau**  
Burp  
Chiffrement SSL

GPU API  
RenderDoc  
CE Autoassembler  
 **Hooking**  
Injection de DLL  
Frida  
Unity Explorer

Edit registre  
Slowdown système  
 **modifs système**  
Changement horloge  
Edit fichiers sauvegarde

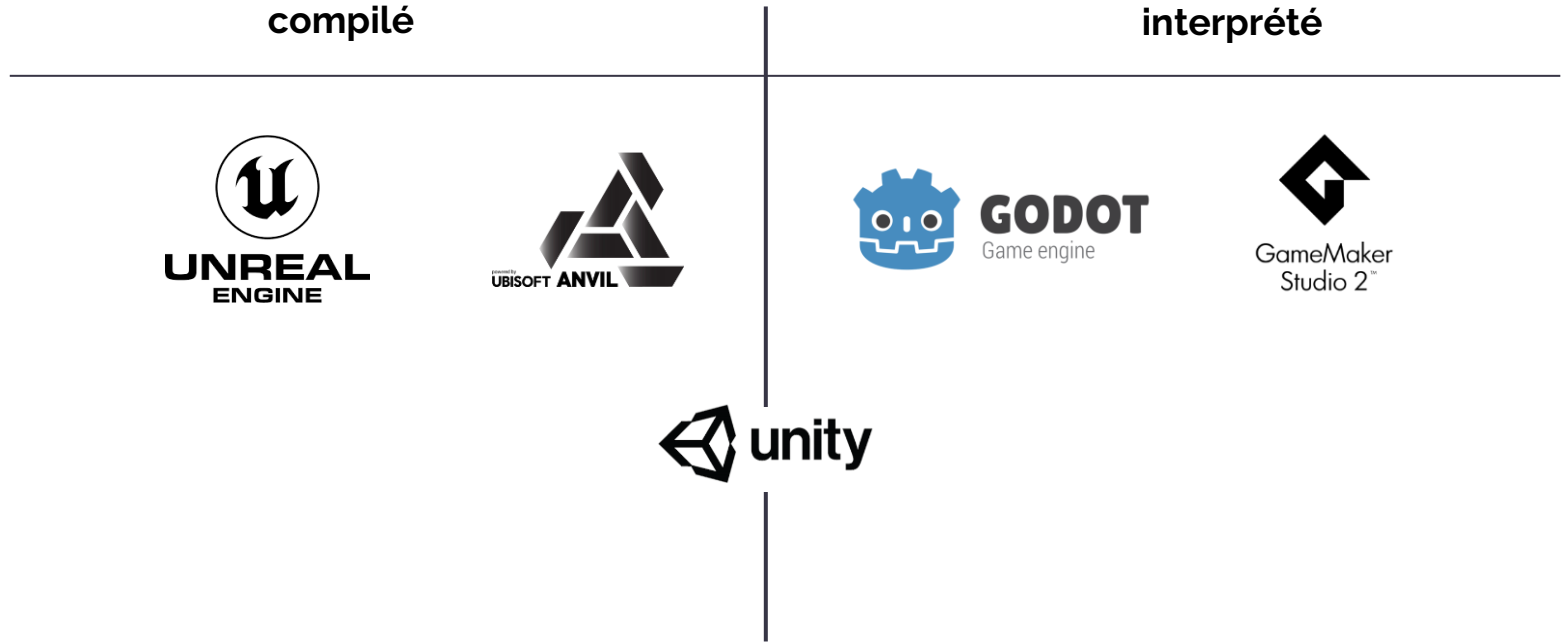
# Différentes méthodes de hack de jeux

## 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



```
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

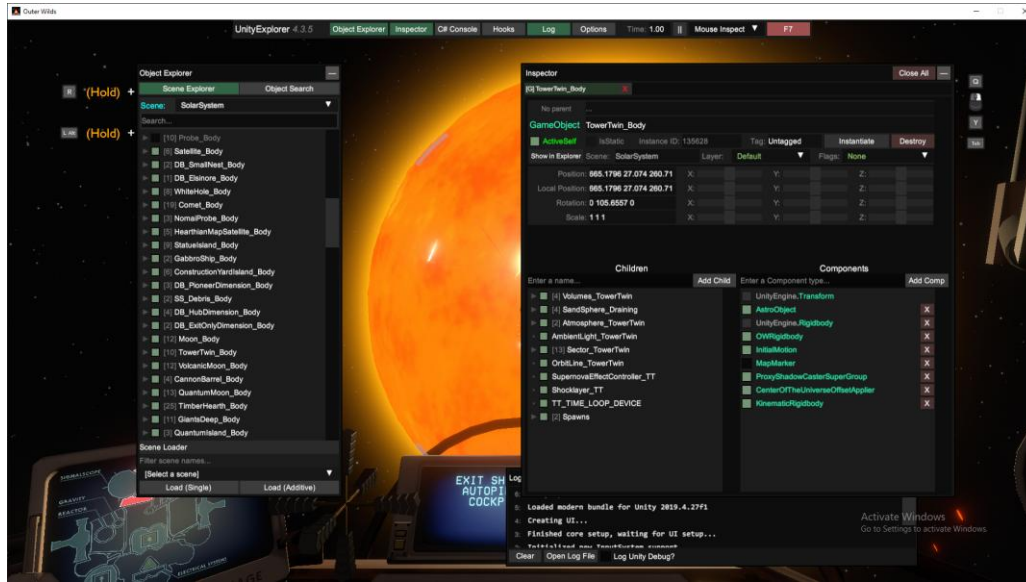
## Hooking de fonctions



Moteur de jeu



process



# Différentes méthodes de hack de jeux

## Hooking de fonctions



Moteur de jeu



Render GPU



process



The screenshot displays the RenderDoc interface with several panels. The top-left panel shows a list of hooks for the game engine, including:

Hook Name	Filter	Load Range	Load Size
Texture 2D	None	0x00000000-0x00000000	0x00000000
Texture 2D	None	0x00000000-0x00000000	0x00000000
Texture 2D	None	0x00000000-0x00000000	0x00000000

The middle-left panel shows the CPU Disassembler with assembly code for a hook function:

```
0 4  [0] mov rax, [0]
1 5  [1] mov rax, [0]
2 6  [2] mov rax, [0]
3 7  [3] mov rax, [0]
4 8  [4] mov rax, [0]
5 9  [5] mov rax, [0]
6 10 [6] mov rax, [0]
7 11 [7] mov rax, [0]
8 12 [8] mov rax, [0]
9 13 [9] mov rax, [0]
10 14 [10] mov rax, [0]
```

The middle-right panel shows the GPU Disassembler with assembly code for a hook function:

```
0 0  [0] mov rax, [0]
1 1  [1] mov rax, [0]
2 2  [2] mov rax, [0]
3 3  [3] mov rax, [0]
4 4  [4] mov rax, [0]
5 5  [5] mov rax, [0]
6 6  [6] mov rax, [0]
7 7  [7] mov rax, [0]
8 8  [8] mov rax, [0]
9 9  [9] mov rax, [0]
10 10 [10] mov rax, [0]
11 11 [11] mov rax, [0]
12 12 [12] mov rax, [0]
13 13 [13] mov rax, [0]
14 14 [14] mov rax, [0]
```

The right panel shows the GPU Viewport with a 3D scene of a character in a Santa hat. The bottom-left panel shows the Shader Editor with a shader graph for a hook function:

```
13 layout(binding = 2) uniform sampler2D sampler;
14 layout(location = 0) float in [in_float];
15 layout(location = 1) int in [in_int];
16 layout(location = 2) float in [in_float2];
17 layout(location = 3) float in [in_float3];
18 void main()
19 {
20     bool flagDetected = true;
21     for (int i = 0; i < in_int; i++)
22     {
23         float in_float = in_float[i];
24         if (in_float > 1.0)
25             flagDetected = false;
26     }
27     float out = in_float;
28 }
```

The bottom-right panel shows the Shader Editor with a shader graph for a hook function:

```
13 layout(binding = 2) uniform sampler2D sampler;
14 layout(location = 0) float in [in_float];
15 layout(location = 1) int in [in_int];
16 layout(location = 2) float in [in_float2];
17 layout(location = 3) float in [in_float3];
18 void main()
19 {
20     bool flagDetected = true;
21     for (int i = 0; i < in_int; i++)
22     {
23         float in_float = in_float[i];
24         if (in_float > 1.0)
25             flagDetected = false;
26     }
27     float out = in_float;
28 }
```

## Différentes méthodes de hack de jeux

 Edit de fichiers de sauvegarde et config

```
TricksData.ini ↗ ✕  
#TRICK_OLLIE  
#Input UP  
tricks.001.tricksName=Ollie  
tricks.001.animation=8  
tricks.001.scoreModifier=100  
Ollie 197K+  
x1 PERFECT! 197,568
```

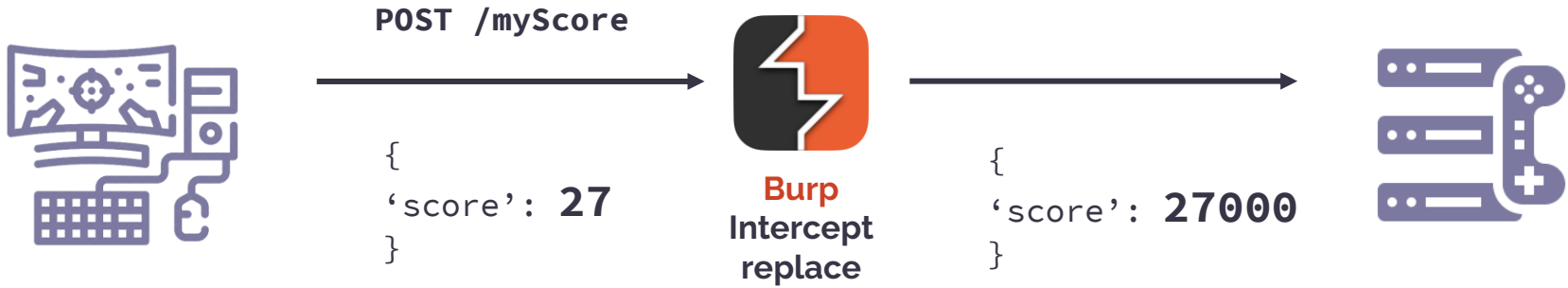


### Procmon64

Trouve les fichiers/  
Clés registres  
accédés par  
un process

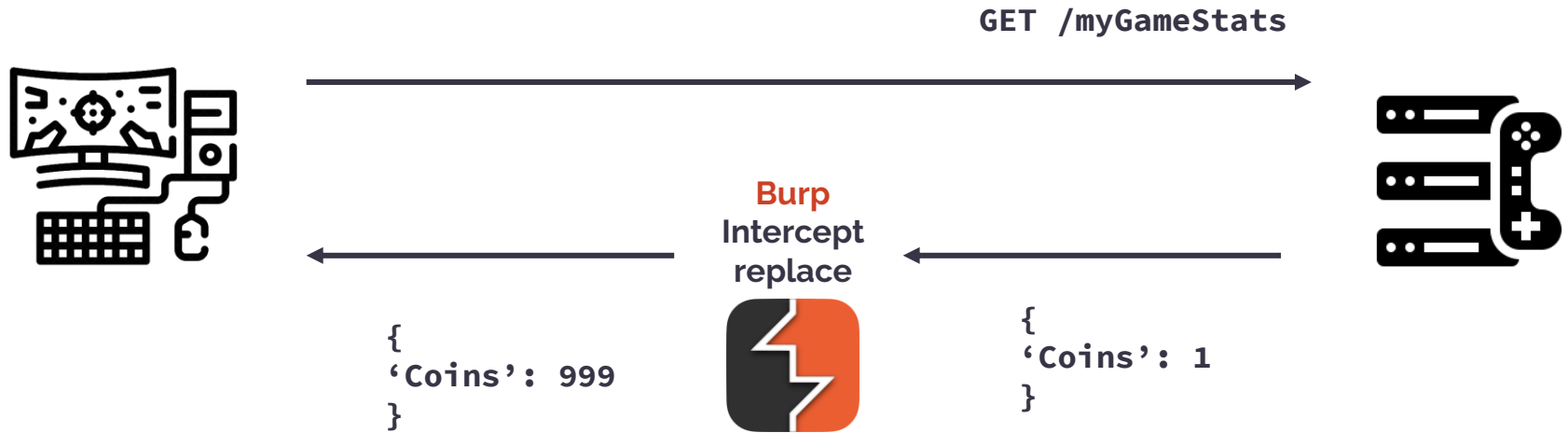
# Différentes méthodes de hack de jeux

## 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”*



- Scanner mémoire et debugger

Cheat Engine 7.5 interface showing a memory scan for the process 0000674-flag\_quest\_release.exe. The scan results are displayed in a table with columns for Address, Value, Previous, and First. The right panel shows scan options including Scan Type (Increased value), Value Type (Float), and Memory Scan Options (All, Start, Stop, Writable, CopyOnWrite, Active memory only, Fast Scan, Alignment, Last Digits, Pause the game while scanning).

Address	Value	Previous	First
E88A9FF35C	55.5	55.5	-0.00003051...
E88B9FF2DC	255.2008057	255.2008057	290.9992065
E88B9FF2F4	255.2008057	255.2008057	290.9992065
E88B9FF30C	55.08114624	55.08114624	-0.00003051...
E88B9FF32C	255.2008057	255.2008057	290.9992065
E88B9FF39C	255.2008057	255.2008057	290.9992065
E88B9FF6D8	0	0	0
E88D9FF2E0	62.46905518	62.46905518	5.99999987214
E88E9FF260	27.27392578	27.27392578	88.62145996
E88E9FF334	256.563446	256.563446	290.9993094
E88E9FF34C	55.08119202	55.08119202	-0.00003051...
E88E9FF36C	256.563446	256.563446	290.9993094
E88E9FF3DC	256.563446	256.563446	290.9993094
E88F1FF850	45.99002075	45.99002075	5.99999987214
E88F1FF8BC	262.0027161	262.0027161	290.9993094
E88F1FF8D4	262.0027161	262.0027161	362
E88F1FF90C	257.9138489	257.9138489	362
E88F1FF97C	257.9138489	257.9138489	362
E88F9FF3D0	36.50170898	36.50170898	89.84651611

Active	Description	Address	Type	Value
<input type="checkbox"/>	No description	28ADC91EE0	4 Bytes	5259
<input type="checkbox"/>	No description	28ADC7B33FC	Float	2064
<input type="checkbox"/>	No description	28ADC7EC544	Float	0
<input type="checkbox"/>	No description	28ADCBE8410	Float	2.087934712E-43
<input type="checkbox"/>	No description	28ADCDE9528	Float	4.593582483E-40
<input type="checkbox"/>	No description	28ADCDE96A0	Float	8.407790786E-45
<input type="checkbox"/>	No description	28ADCFC090	Float	-1.892436462E33
<input type="checkbox"/>	No description	28ADC6998C0	Float	Nan
<input type="checkbox"/>	No description	28ADC7B3250	Float	0
<input type="checkbox"/>	No description	28ADD2C7CB8	4 Bytes	100004
<input type="checkbox"/>	No description	28ADD2D17F8	4 Bytes	1000000
<input type="checkbox"/>	No description	28AF6EB4040	4 Bytes	999992

# 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
```



- 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 -- disable
    local addr = getAddress(symbol)
    autoAssemble(script, disableInfo) -- disable script and
    return addr, 'success'
end

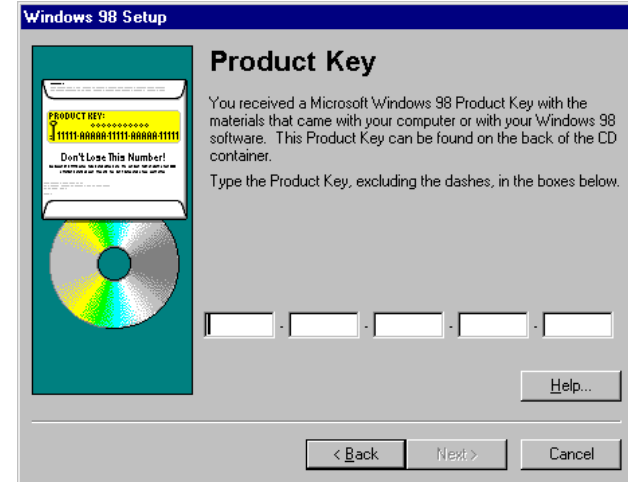
function AOBScanRegion(bytestr, start, stop)
    local script = ([[
    [ENABLE]
    aobscanregion(luaAOBScanRegionSymbol,%X,%X,%s)
    registersymbol(luaAOBScanRegionSymbol)
    [DISABLE]
    unregistersymbol(luaAOBScanRegionSymbol)
    ]]):format(getAddress(start), getAddress(stop), bytestr)
    return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

- Scanner mémoire et debugger
- scripting AutoAssembly et LUA
- Générateur de GUI 'trainer'



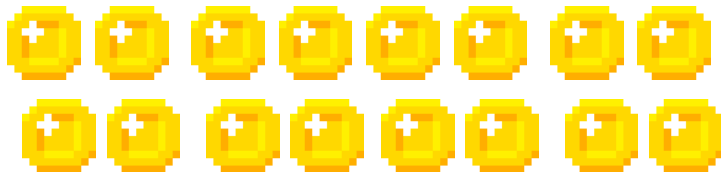
- Scanner mémoire et debugger
- scripting AutoAssembly et LUA
- Générateur de GUI 'trainer'
- **Pas limité au hacking de jeux**





# Trouver des valeurs

COINS : 4



New Scan Next Scan

Value:

Hex

Scan Type

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
13193...	4	4
15C0A...	4	4

## Trouver des valeurs

COINS : 20



New Scan Next Scan

Value:

Hex 20

Found:4

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



## Trouver des valeurs

COINS : 20



New Scan Next Scan

Value:

Hex 20

Active	Description	Address	Type	Value
<input type="checkbox"/>	coins	00000000		
<input type="checkbox"/>	coins	15C0DD859784	Bytes	20
<input type="checkbox"/>	coins	15C0DD859AC4	Bytes	20
<input type="checkbox"/>	coins	15C0EC83FD84	4 Bytes	20
<input type="checkbox"/>	coins	15C0DD859C84	Bytes	20

Change Value

what value to change this to?

4000

OK

## Trouver des valeurs

COINS : 4000



Active	Description	Address	Type	Value
<input type="checkbox"/>	coins	00000000		
<input type="checkbox"/>	coins	15C0DD859784	Bytes	20
<input type="checkbox"/>	coins	15C0DD859AC4	Bytes	20
<input type="checkbox"/>	coins	15C0EC83FD84	Bytes	20
<input type="checkbox"/>	coins	15C0DD859C84	Bytes	20

Change Value

what value to change this to?

4000

OK

## Trouver des valeurs: features avancées

The image displays a memory scanner interface with several overlapping panels. The central panel shows a search configuration:

- Value:** `value % 2 == 1 and value > previousvalue * 3`
- Lua formula**
- Hex** 20
- Scan Type:** Exact Value
- Value Type:** 4 Bytes
- Lua formula**
- Not**
- Memory Scan Options:**
  - All** (selected)
  - Start:** 0000000000000000
  - Stop:** 00007fffffff
  - Writable**
  - Executable**
  - CopyOnWrite**
  - Active memory only** (with a red 'x' icon)
  - Fast Scan** (with a box containing '4')
  - Alignment**
  - Last Digits**
  - Pause the game while scanning**

Two side panels show dropdown menus:

- Left Panel:** A list of search criteria including "Exact Value", "Bigger than...", "Smaller than...", "Value between...", "Increased value", "Increased value by ...", "Decreased value", "Decreased value by ...", "Changed value", "Unchanged value", and "Ignore value". "Exact Value" is selected.
- Right Panel:** A list of data types including "4 Bytes", "Binary", "Byte", "2 Bytes", "4 Bytes", "8 Bytes", "Float", "Double", "String", "Array of byte", "All", and "Grouped". "4 Bytes" is selected.

# 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

Memory View    Display Type > • 4 Byte decimal

Protect:Read/Write    AllocationBase=15C0DD4000 Base=


address	'78	7C	89ABCDEF	01234567
15C0DD85978	1337	100	9...d...	... \...
15C0DD85988	200	22	.....	.....
15C0DD85998	694768000	348	Qi) \...	.....
15C0DD859A8	242510512	348	jt. \...	.....
15C0DD859B8	6086	8192	....	Qm) \...
15C0DD859C8	14099	256	.7.....	B.. \...

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

# Data structures

Description	Address	Type	Value
health	15C0DD85978	4 Bytes	1337

Memory View

Tools  Dissect data/structures

Offset-description | Address: Value

Player

```
...0000 - 4 Bytes 5DA650 : 1337
...0004 - 4 Bytes 5DA654 : 100
...0008 - 4 Bytes 5DA658 : 200
...000C - 4 Bytes 5DA65C : 22
```

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

## Adresses mémoire persistantes.

Activ	Description	Address	Type	Value
<input type="checkbox"/>	coins	00000000		
<input type="checkbox"/>	coins	15C0DD859784	Bytes	??
<input type="checkbox"/>	coins	15C0DD859AC4	Bytes	??
<input type="checkbox"/>	coins	15C0EC83FD84	4 Bytes	??
<input type="checkbox"/>	coins	15C0DD859C84	Bytes	??



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.

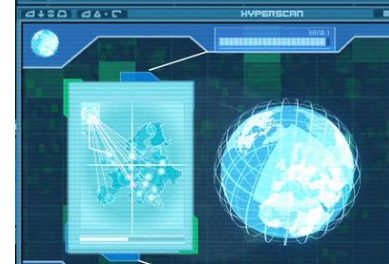
solution: rescan, et comparer les résultats

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

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**



Addresses executed since last filter operation: 1

Has been executed

Has not been executed

Address List (1)

Address	Executed
Tutorial-i386.exe.text+1BB5	Yes

```
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 ==

---

**jne** if !=

**jg** if >

---

**jg** if <

**add** +=

---

**sub** -=

**mov** x=y

**nop** do nothing

(padding)

## 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*

```
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*

```
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
```

```
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?

coins      01723548      4 Bytes   100      Find out what writes to this address

The following opcodes write to 01723548

Count	Instruction
1	004272D7 - 89 02 - mov [edx],eax

\*\*\*\*\*

Tutorial-i386.exe.text+262D7:  
004272CE - 8B 15 B0666500 - mov edx,[Tutorial-i386.exe+2566B0]  
004272D4 - 8B 45 F0 - mov eax,[ebp-10]  
004272D7 - 89 02 - mov [edx],eax <<

EAX=0000037F  
EBX=00000000

Replace  
Show disassembler  
Add to the codelist  
More information  
copy memory

## Instruction patching: problèmes

### Instructions moins évidentes

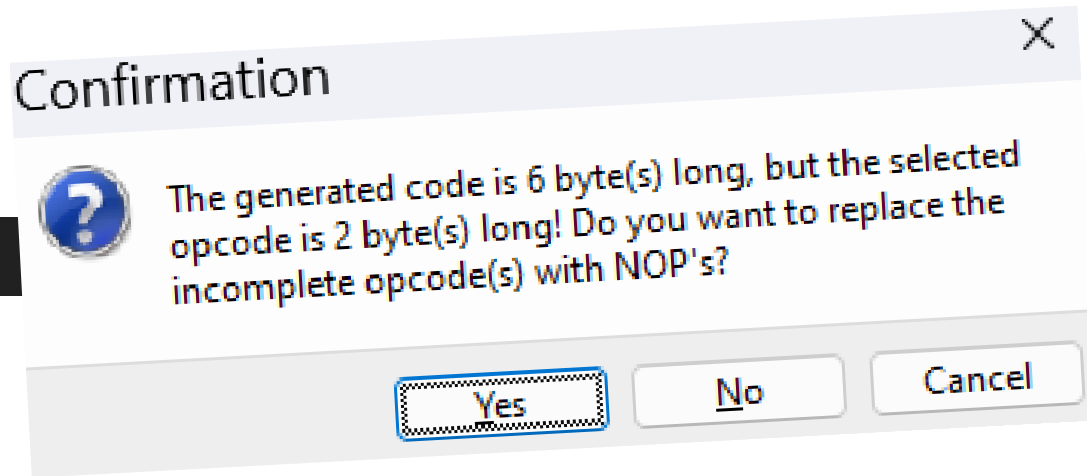
```
mov [edx], eax
```

where **add** ?



## Instruction patching: problèmes

### Clash de tailles d'instructions





## Instruction patching: problèmes

### Clash de tailles d'instructions

```
mov [edx],00001000
```

```
mov [edx],0000100036.exe+2566B0]
```

## Instruction patching: problèmes

Ou rajouter notre code?

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

## Instruction patching: problèmes

Ou rajouter notre code?

```
mov     eax,[ebp-10]
mov     [edx],eax
add     eax,[02torial-i386.exe+2566B0]
mov     [edx],eax
```

# 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  
gère labels, variables...

<input type="checkbox"/>	multiplier	018E0800	4 Bytes	5
--------------------------	------------	----------	---------	---

```
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
```

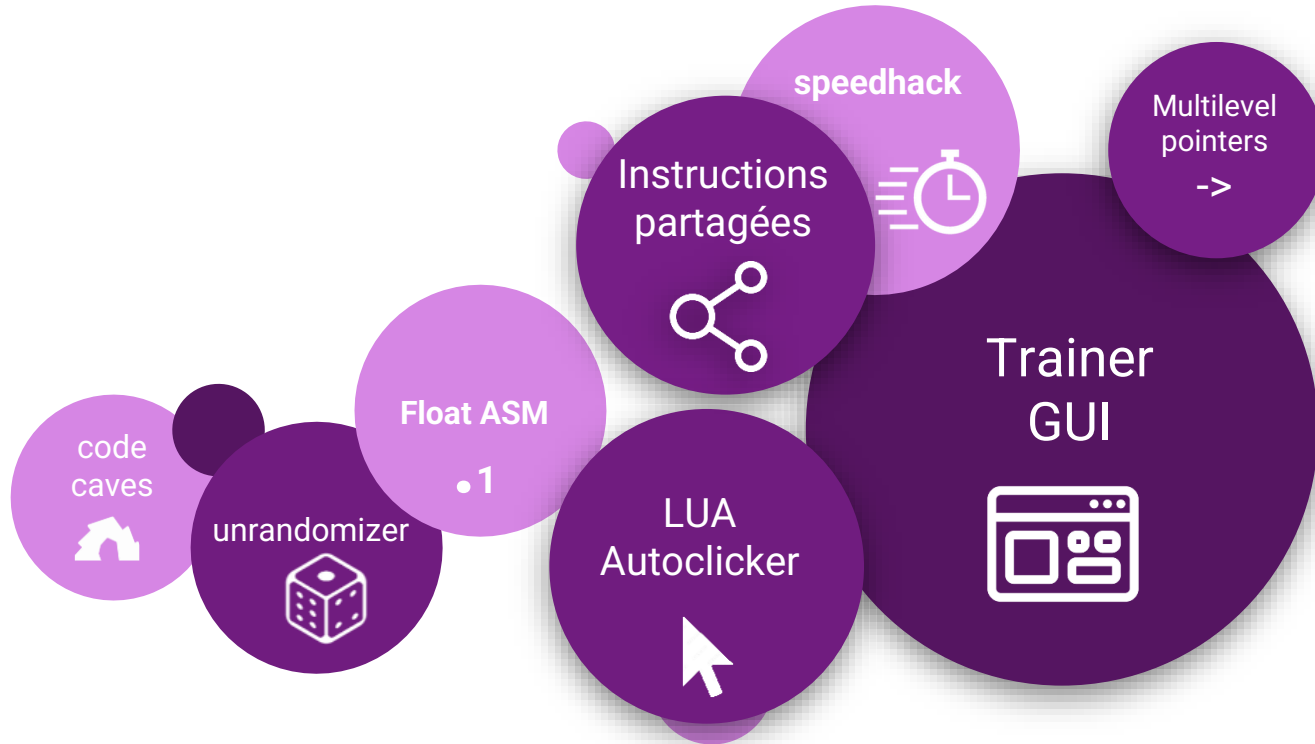
```
INJECT:
jmp newmem
nop 2
return:
```

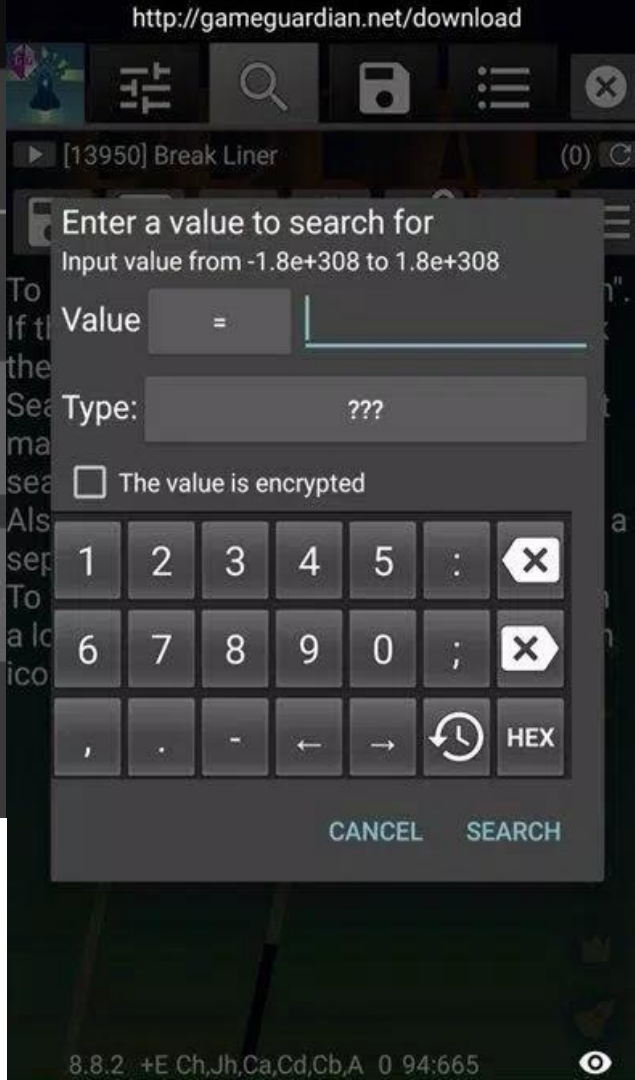
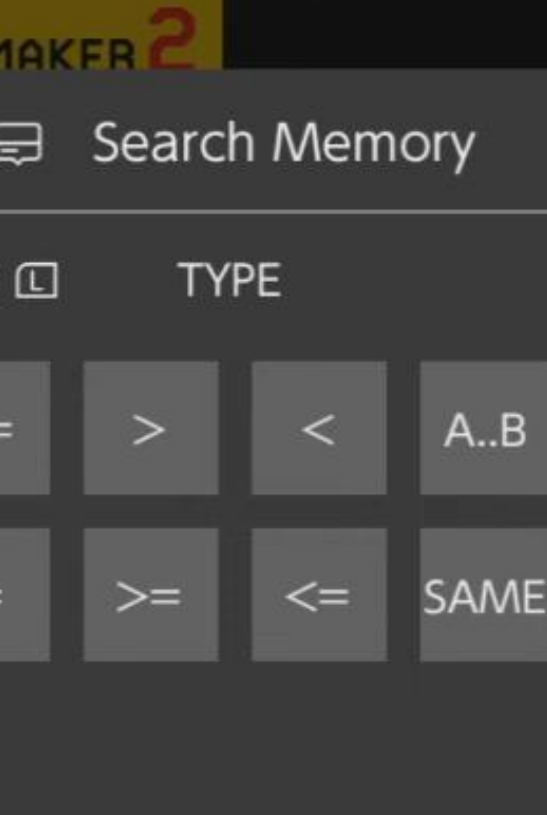
```
[DISABLE]
dealloc(multiplier)
unregisterSymbol(multiplier)
```

```
dealloc(newmem)
```

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

# Next steps







Equivalents sur toutes plateformes

Plus compliqués à setup

# Resources

- Forums Cheat Engine

Author	Post
<b>tuxlu</b> How do I cheat? Reputation: 0 Joined: 24 Sep 2023 Posts: 3	<p>Posted: Sun Sep 24, 2023 2:05 pm Post subject: need explain</p> <p>Hi!</p> <p>I'm doing a presentation on Cheat Engine and now I wa</p> <p>So for pointer maps, is this stackoverflow post "About</p> <p>summarised, it says pointer map searches recursively</p> <p><b>Code:</b></p> <pre>add [(addr-08) + 08, 42]</pre> <p>what I don't understand really, is that dynamic debuggi</p> <p>what obvious thing did I miss?</p> <p>I know Guided Hacking has a detailed article on this, b</p> <p><a href="#">Back to top</a> <a href="#">profile</a> <a href="#">pm</a></p>
<b>ParkourPenguin</b> I post too much  Reputation: 127 Joined: 06 Jul 2014 Posts: 3924	<p>Posted: Sun Sep 24, 2023 3:05 pm Post subject:</p> <p>Addresses don't get accessed if the code that accesses</p> <p>The overwhelming majority (&gt;99.99%) of the pointer p</p> <p>Basically, the pointer scanner can be dumbed down into</p> <ol style="list-style-type: none"><li>1. Address is given to the pointer scanner</li><li>2. Scan for pointer values between (address - max_off</li><li>3. For each result, go back to step 1</li></ol> <p>There's lots of other small details</p> <p>I don't know where I'm going, but I'll figure it out when</p> <p><a href="#">Back to top</a> <a href="#">profile</a> <a href="#">pm</a></p>
<b>Dark Byte</b> Site Admin  Reputation: 452 Joined: 09 May 2003 Posts: 25009 Location: The netherlands	<p>Posted: Mon Sep 25, 2023 12:27 am Post subject:</p> <p>That's why it's important to have a 2nd pointermap fro</p> <p>Do not ask me about online cheats. I don't know any and wont help finding them</p> <p>Like my help? Join me on <a href="#">Patreon</a> so I can keep helping</p> <p><a href="#">Back to top</a> <a href="#">profile</a> <a href="#">pm</a> <a href="#">msnm</a> <a href="#">ICQ</a></p>

# 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



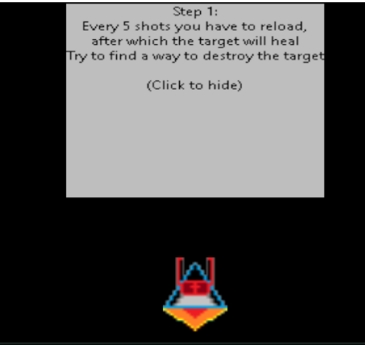
Pwn Adventure



**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



devoteam  
Cyber Trust

trailer music: Robyn - touch (edited)

# Godot Engine



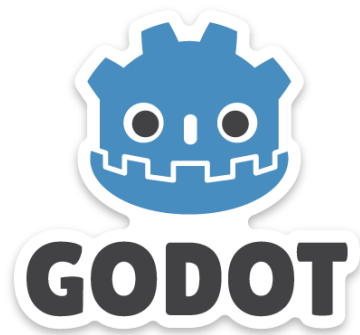
Moteur libre et open source



Léger et iteration rapide



Communauté grandissante



Pas encore là pour la grosse 3D

Performances améliorables

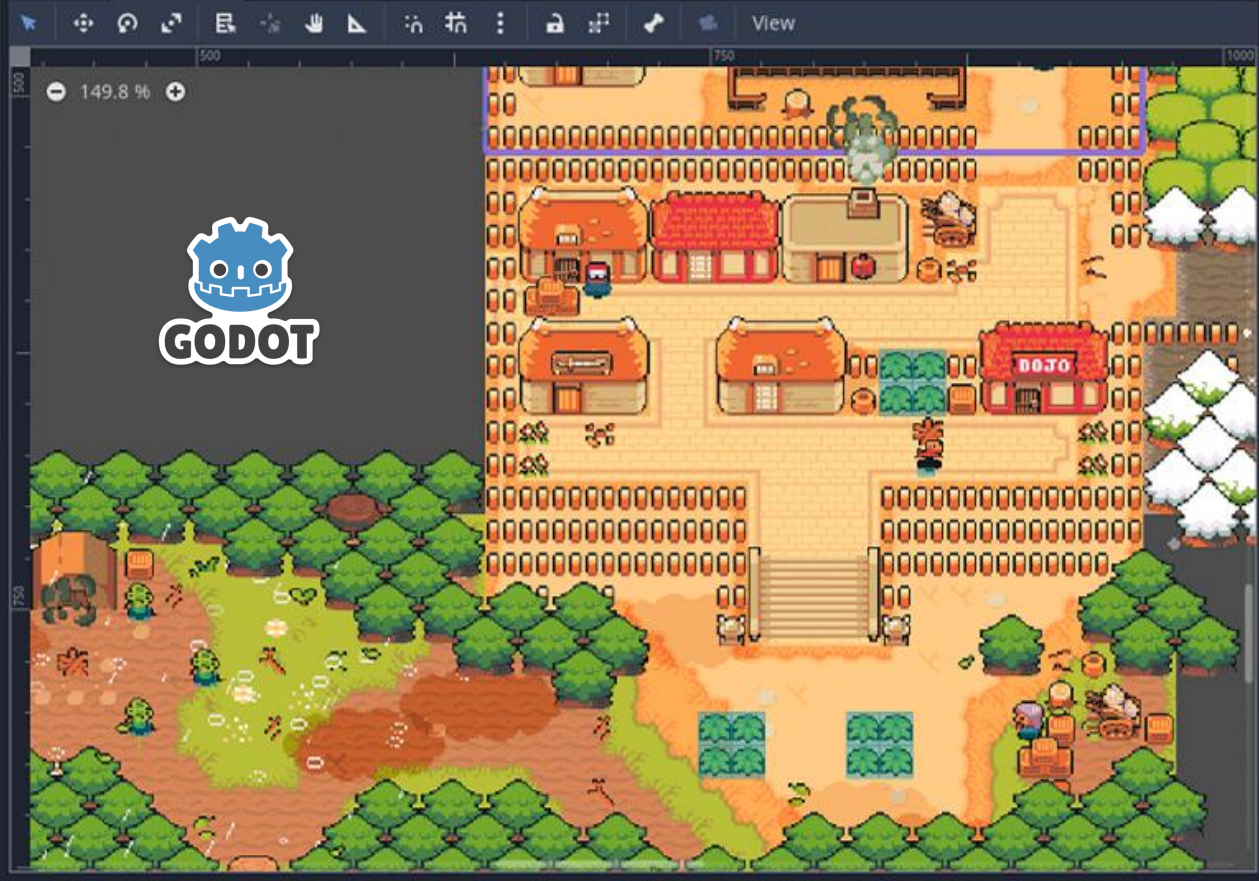
Manque de features avancées



Scene Import

Main World

- 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

Inspector Node

- PostProcessing
- Filter properties
- WorldEnvironment
    - Environment
      - 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



Scene Import

Filter Nodes

- Main
  - Player
    - AnimatedSprite2D
    - CollisionShape2D
    - AudioStreamPlayer2D
    - Camera2D
  - Enemy
    - AnimatedSprite2D
    - CollisionShape2D
  - Area2D
    - CollisionShape2D

[empty] X +

File Edit Search Go To Debug

Filter Scripts

Player.gd

```

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

```

FileSystem

res://

Search files

- res://
  - Hud
  - Main
  - Menu
  - Resource
  - World
  - default\_bus\_layout.tres
  - default\_env.tres
  - Icon.png

Player.gd

Filter Methods

\_physics\_process

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

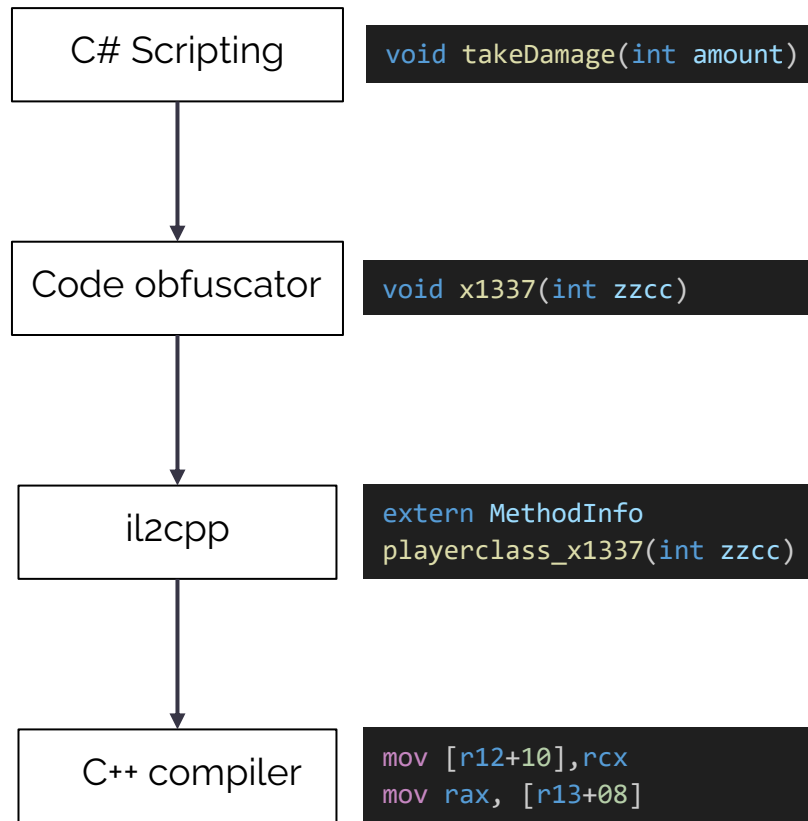
- Obfusquer les valeurs en mémoire

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

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

# Protéger votre jeu

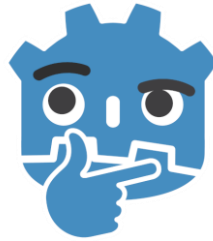
- Obfusquer les valeurs en mémoire
- **Obfusquer les binaires**



# Protéger votre jeu: Godot



Langage interprété et  
format open source

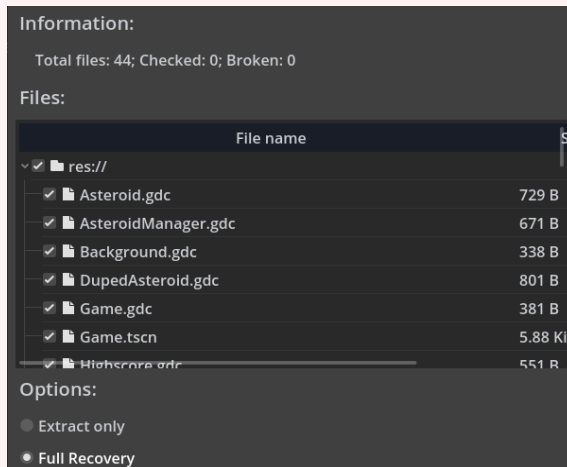


**Kalm**

# Protéger votre jeu: Godot



Langage interprété et  
format open source



**Décompilable entièrement  
avec GdsDecomp**



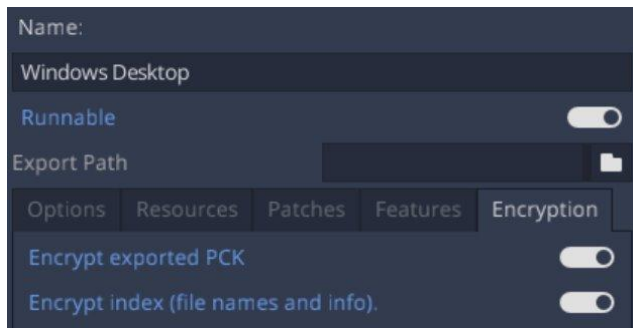
**Panik**

# Protéger votre jeu: Godot



Langage interprété et  
format open source

**Le jeu peut être chiffré  
avec une clé AES**



Décompilable entièrement  
avec GdsDecomp



**Kalm**

# Protéger votre jeu: Godot



Langage interprété et  
format open source

Le jeu peut être chiffré  
avec une clé AES

```
; Compare Two Operands
847D7 ; Jump if Greater or Equal (SF=OF)

lea rcx, [rbp+190h+p_key._cowdata] ; this
call ?_copy_on_write@?$CowData@D@@@AEAAIXZ ;
mov rcx, [rbp+190h+p_key._cowdata_ptr]
lea rax, ?script_encryption_key@@@3PAEA ; uc
movzx eax, byte ptr [rbx+rax] ; Move with Zer
mov [rcx+rbx], al
```



Panik



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**

nikitalita commented on Jul 23, 2022

you can use IDA to get the decryption key.

Originally, specific steps were provided, but after careful consideration, it may affect the enthusiasm of Godot developers, so the specific practice was deleted



Décompilable entièrement  
avec GdsDecomp

La clé est extractable  
dans le binaire



**Kalm**

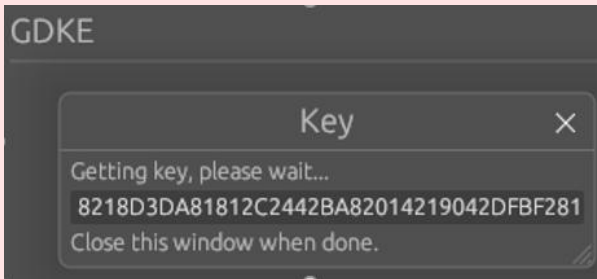
# Protéger votre jeu: Godot



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**



Décompilable entièrement  
avec GdsDecomp

La clé est extractable  
dans le binaire

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

```
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>());
```



**Kalm**

# 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

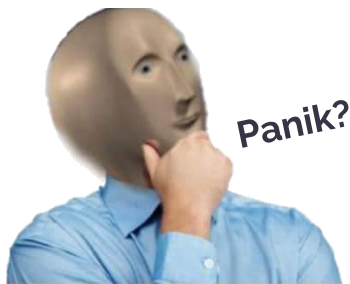
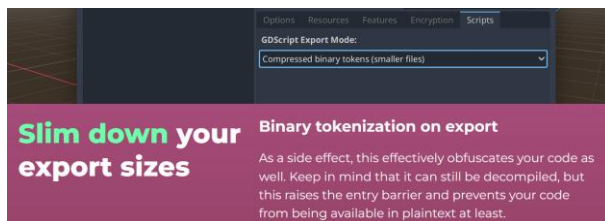
On peut modifier  
quelques lignes du  
moteur pour fool le tool



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**

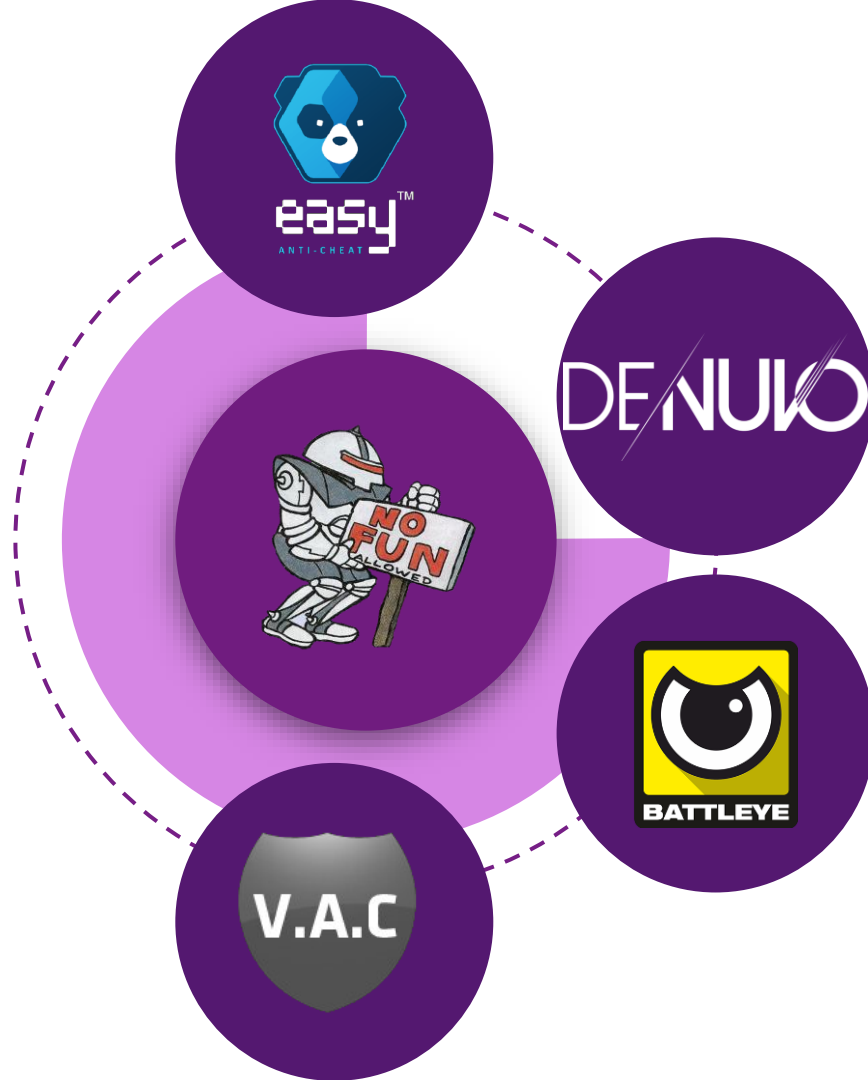


# Protéger votre jeu

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

**DISCREPANCY  
DETECTED**

# Logiciels Anti cheat



# Logiciels Anti cheat

Du Driver Windows à l'EDR - Aurelien Chalot



Analyse tous  
les process



Détecte  
hook fonctions



En gros



Check  
mémoire



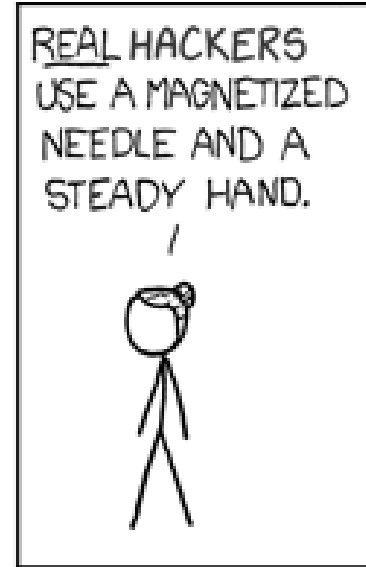
driver  
Kernel mode





## Advanced cheating: Sans toucher le jeu.

- Hacks hardware,  
plus difficile à détecter.



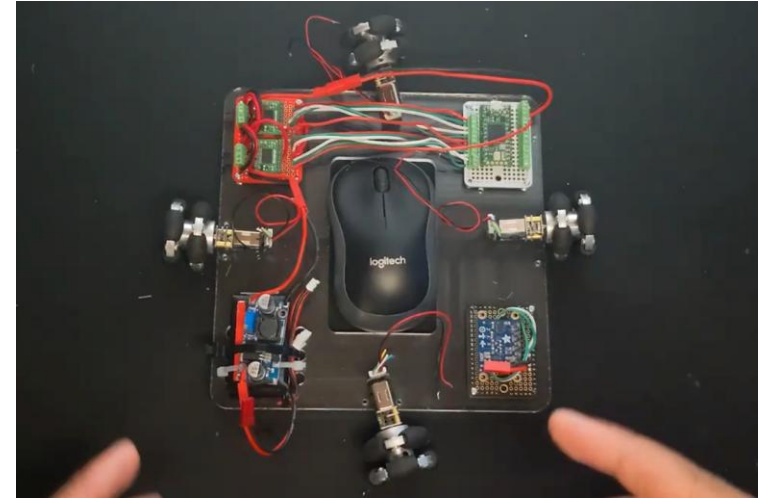
## Advanced cheating: Sans toucher le jeu.

- 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**



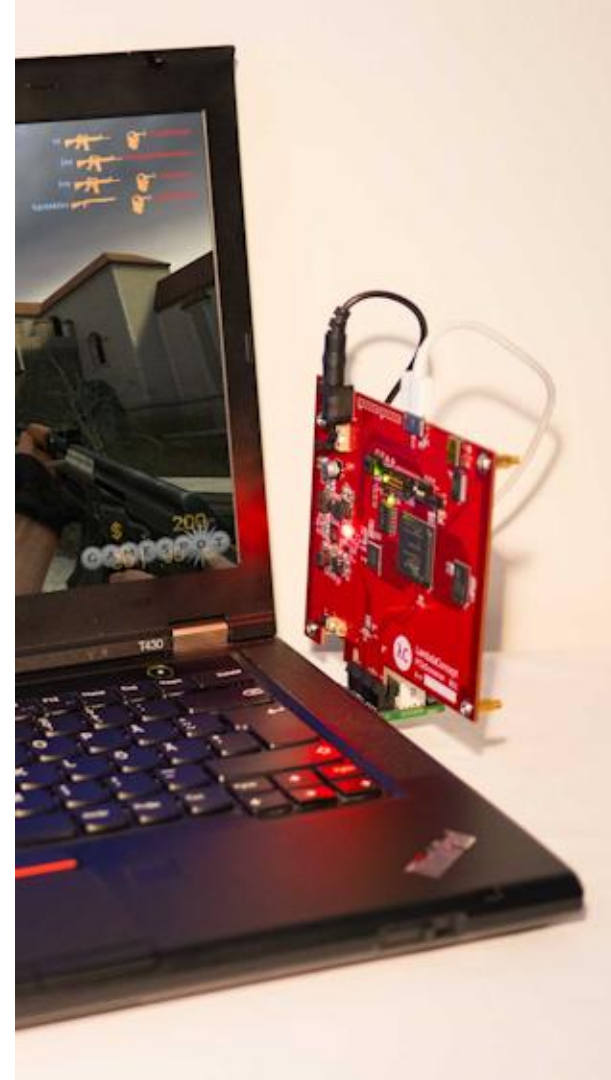
## Advanced cheating: Sans toucher le jeu.

- 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**



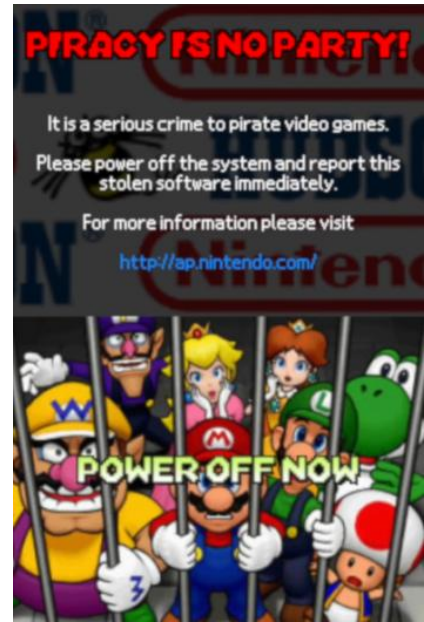


[tuxlu.fr/talk\\_vghacking](http://tuxlu.fr/talk_vghacking)





[tuxlu.fr/talk\\_vghacking](http://tuxlu.fr/talk_vghacking)







thank you.

