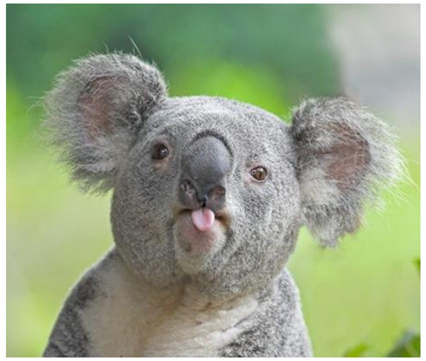
DIMITRI FOURNY
Security Researcher / dfourny@quarkslab.com

Attacking Games for Fun and Profit

quarkslab
SECURING EVERY BIT OF YOUR DATA

Who Am I

- Dimitri Fourny (@DimitriFourny / dimitrifourny.com)
- Student at Polytech'Lille (until this evening \o/)
- Security Researcher at Quarkslab
- Reverse Engineering / Fuzzing / Exploitation



Attacking Games

In the past, I have reversed:

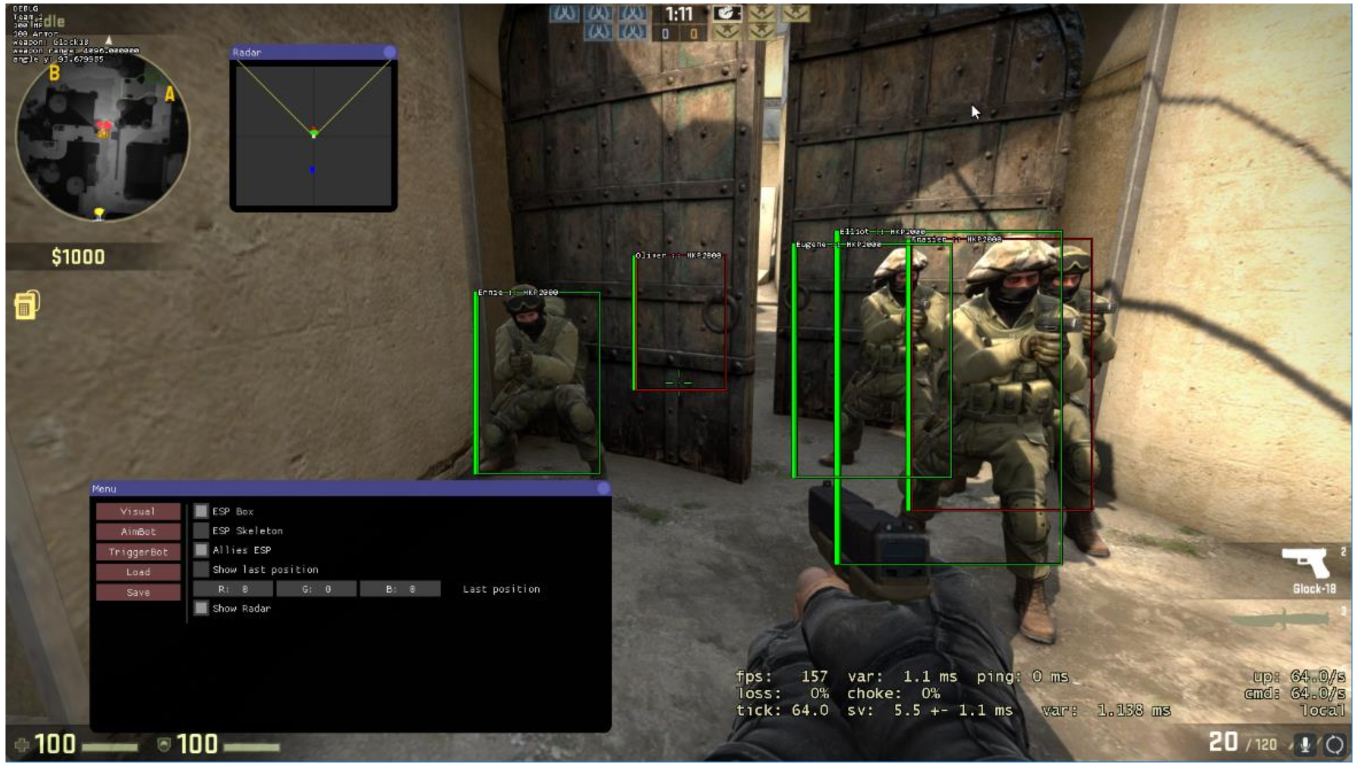
- Counter Strike Source
- Team Fortress 2
- Battlefield 3
- Battlefield 4 (Frostbite Engine)
- League of Legends
- **Counter Strike Global Offensive (Source Engine 2)**

For Fun and Profit

- To see how the game works
- To create amazing mods
  - Ex: GTA San Andreas Multiplayer mod
- To make cheats
  - Aim assist
  - View enemy through walls
- For profit: the most known cheats make more than $500.000 per year

Cheats Functionalities

- Visuals
  - ESP
  - Chams
  - Radar hack
  - No smoke / No flash
- AIM assist
  - AIM bot
  - Trigger bot
  - Silent AIM
  - Auto fire



ESP Box
ESP Skeleton
Allies ESP
Show last position
R: 0 G: 0 B: 0 Last position
Show Radar

```
fps:   157  var:  1.1 ms  ping: 0 ms                      up: 64.0/s
loss:   0%  choke:  0%                                   cmd: 64.0/s
tick: 64.0  sv:  5.5 +- 1.1 ms   var:  1.138 ms               local
```

Injection

- How to modify the game memory?
- External hack
  - OpenProcess()
  - ReadProcessMemory() / WriteProcessMemory()
- Internal hack: Inject a DLL in the process csgo.exe
  - OpenProcess()
  - CreateRemoteThread(lpStartAddress = @LoadLibrary, lpParameter = "cheat.dll")

Dynamic Link Library

DLL	Description
Client.dll	Basic movement game loop (CreateMove()) and Network Data Tables
VguiMatSurface.dll	Contain the basic 2D functions (ex: drawLine(), drawPrintText())
Vgui2.dll	Contain the game loop for the interface
Engine.dll	Basic functions from the CS:GO engine (getLocalPlayer(), worldToScreenMatrix)
InputSystem.dll	Cursor and keyboard functions
ShaderApiDx9.dll	Contain the pointer to the IDirect3DDevice9

Searching the Class Names

- Locate client.dll (*Steam\steamapps\common\Counter-Strike Global Offensive\csgo\bin*)
- Open client.dll in IDA
- *Views -> Open subviews -> Strings*
- We found *'VClient018'* in .rdata => CS:GO has been updated!
- Repeat the process for all class/dll

Interface Loading

If you want the class *VClient017* in client.dll:

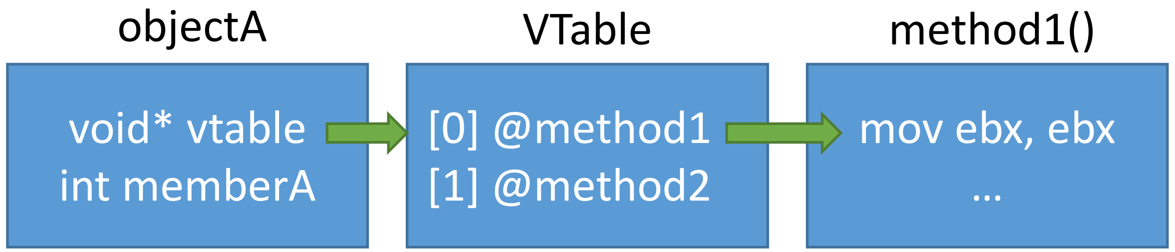
Client->CreateInterface(‘*VClient017*’, 0);

```
HMODULE hModule = GetModuleHandle("client.dll");
CreateInterfaceFn CreateInterface = GetProcAddress(hModule, "CreateInterface");
Client* client = CreateInterface("VClient017", 0);
```

Functions Address

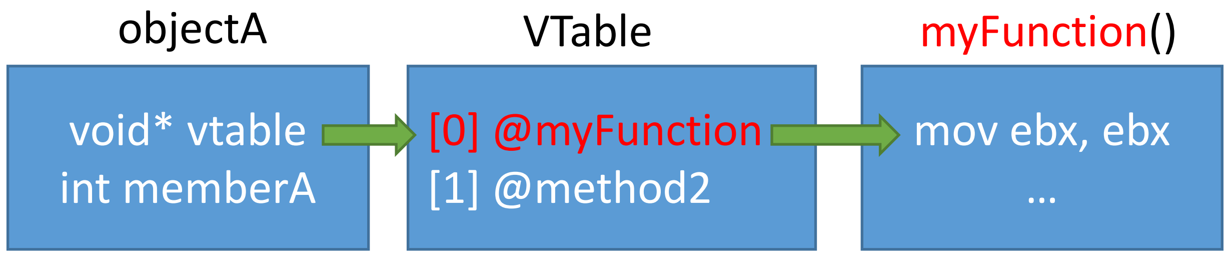
Functions are virtual => Virtual Table

```
class A {
    virtual int method1();
    virtual int method2();
    int memberA;
}
A objectA = new A();
vtable = *(PDWORD) objectA;
```



Hooking

Just replace the pointer to *method1()* with a pointer to *myFunction()*



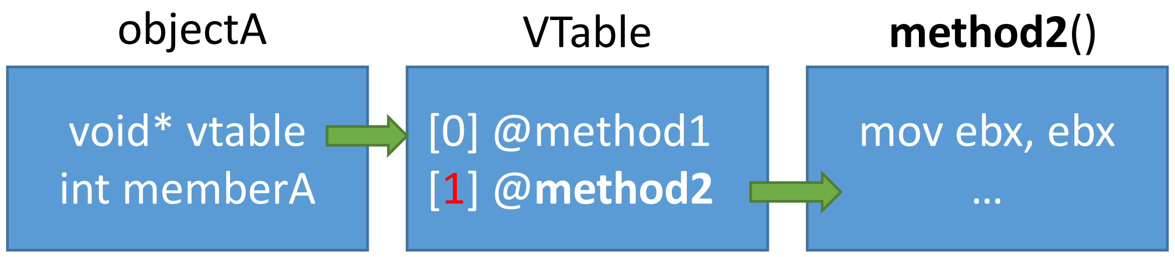
```
A* objectA = new A();
vtable = *(PDWORD) objectA;

vtable[0] = myFunction;  // hook
objectA->method1();      // call vtable[0] => myFunction()
```

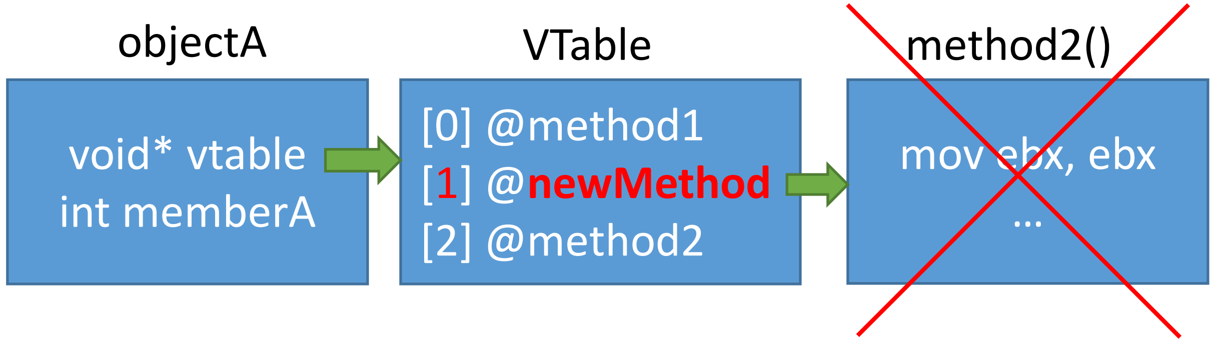
Updating VTable

```
class A {
    virtual int method1();
    virtual int newMethod();
    virtual int method2();
    int memberA;
}
```

Before the update:



After the update:



NetVAR Offsets

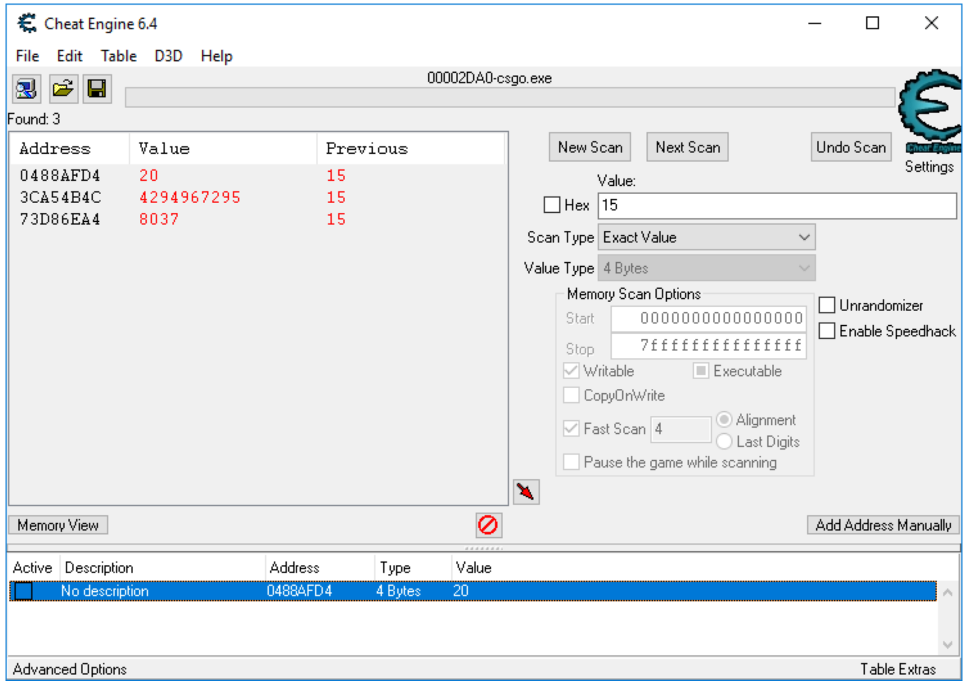
- The Source Engine use Network Data Tables, usually called NetVar
- The table contain relative offsets from the class
- Used for plugin to be "update independent"
- Example:

```
m_health = client->netvarOffset("DT_CSPlayer", "m_iHealth");

int BaseEntity::getHealth() {
	return *(int*)((DWORD)self() + m_health);
}
```

Finding Structures

- Using CheatEngine is always a good start
- The procedure is always the same, for example to find the ammo:
  - Make a first scan with the number of ammo you have
  - Make some shots
  - Search again with the new value
  - Do it again...
- Finally, search for the a static pointer (right click -> Find what read/write to this address)



00002DA0-csgo.exe

Found: 3

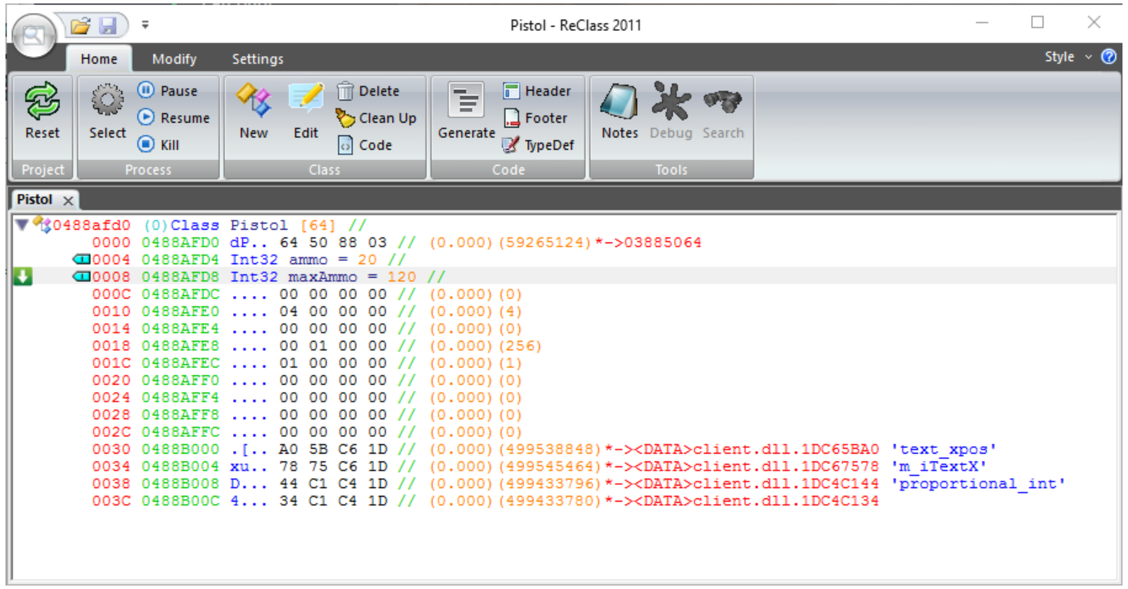
Address	Value	Previous
0488AFD4	20	15
3CA54B4C	4294967295	15
73D86EA4	8037	15

Value: 15
Hex ☐
Scan Type: Exact Value
Value Type: 4 Bytes
Memory Scan Options
Start: 0000000000000000
Stop: 7fffffffffffffff
✓ Writable ▨ Executable
☐ CopyOnWrite
✓ Fast Scan 4 ● Alignment ○ Last Digits
☐ Pause the game while scanning
☐ Unrandomizer
☐ Enable Speedhack

Active	Description	Address	Type	Value
☐	No description	0488AFD4	4 Bytes	20

Finding Complex Structures

- Find one element with CheatEngine
- Open the address in ReClass and reverse the rest of structure
- To help, you can reverse the functions which use this structure in IDA

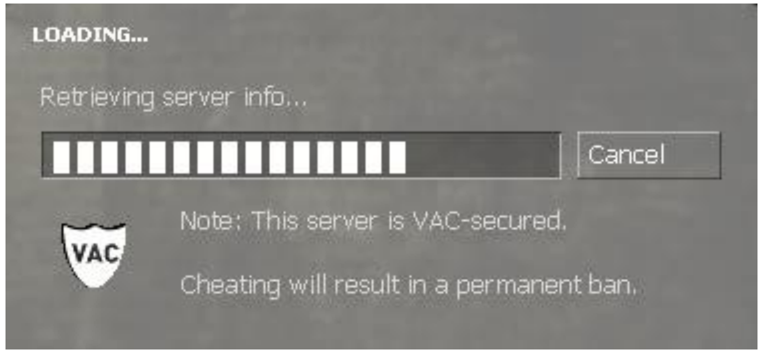


```
0488afd0 (0)Class Pistol [64] //
     0000 0488AFD0 dP.. 64 50 88 03 // (0.000)(59265124)*->03885064
     0004 0488AFD4 Int32 ammo = 20 //
     0008 0488AFD8 Int32 maxAmmo = 120 //
     000C 0488AFDC .... 00 00 00 00 // (0.000)(0)
     0010 0488AFE0 .... 04 00 00 00 // (0.000)(4)
     0014 0488AFE4 .... 00 00 00 00 // (0.000)(0)
     0018 0488AFE8 .... 00 01 00 00 // (0.000)(256)
     001C 0488AFEC .... 01 00 00 00 // (0.000)(1)
     0020 0488AFF0 .... 00 00 00 00 // (0.000)(0)
     0024 0488AFF4 .... 00 00 00 00 // (0.000)(0)
     0028 0488AFF8 .... 00 00 00 00 // (0.000)(0)
     002C 0488AFFC .... 00 00 00 00 // (0.000)(0)
     0030 0488B000 .[.. A0 5B C6 1D // (0.000)(499538848)*-><DATA>client.dll.1DC65BA0 'text_xpos'
     0034 0488B004 xu.. 78 75 C6 1D // (0.000)(499545464)*-><DATA>client.dll.1DC67578 'm_iTextX'
     0038 0488B008 D... 44 C1 C4 1D // (0.000)(499433796)*-><DATA>client.dll.1DC4C144 'proportional_int'
     003C 0488B00C 4... 34 C1 C4 1D // (0.000)(499433780)*-><DATA>client.dll.1DC4C134
```

Anti-Cheats

Most used anti-cheats:

- Valve Anti-Cheat (VAC)
- Punkbuster
- FairFight: server-side, based on statistics


LOADING...

Retrieving server info...

Note: This server is VAC-secured.

Cheating will result in a permanent ban.

VAC3

- Manually mapped in memory
- Detect Windows API trampoline hooks
  - If a hook is detected, it send the module name to the server
- Detect if you have disabled the driver signature verification
  - If it's the case, you will have some deeper verification
- Dump the IP and the MAC address (useful to detect a cheater on a free multiplayer game)
- Detect the injectors using the *Update Sequence Number Journal*, which is a feature of *NTFS*
  - Can be bypassed with an USB key or an Silverlight injector
- Memory signature scans for **public** cheats

Conclusion

- Cheat/Anticheat is the same cat-and-mouse game than malwares/antivirus
- More difficult when you the engine is not public
  - But not always! The Battlefield 3 PDB has been leaked
  - Same thing for Call of Duty 4
- If you want to go deeper: *unknowncheats.me*
- My CS:GO cheat source code will be released soon

quarkslab

SECURING EVERY BIT OF YOUR DATA