Applied Reverse Engineering

Week 0x01: Principles of Game Hacking

Stephen Tong / Fall 2019

Agenda

- What is hacking?
- How can we cheat?
- What we can and can't do (Demo)
- Memory model

What is Game Hacking?

Externally modifying the behavior of the game to effect an advantage

Scripts vs. Hacks

- Scripts are “dumb”
  - Interact with the game only through normally-accessible means (i.e. simulating input, reading pixels, etc.)
  - Doesn’t modify the behavior of the game
- Hacks are “smart”
  - Interact with the game directly (i.e. memory or code execution)
  - Can modify game behavior

Scripts vs. Hacks

Scripts:

- AutoHotKey
- Color-based triggerbots
- Etc.

Hacks:

- Cheat Engine (memory editor)
- "Trainers" (externals hacks)
- "Injectors" (internals)
- Packet editing

Scripts vs. Hacks

Scripts:

- AutoHotKey
- Color-based triggerbots
- Etc.

Hacks:

- Cheat Engine (memory editor)
- "Trainers" (externals hacks)
- "Injectors" (internals)
- Packet editing

Hacks Depend on the Game

- RTS: Maphack, botting
- MMORPG: All Map Attack, Entity Vacuum
- Sandbox: Nuker, Fastbuild
- FPS: Aimbot, Wallhack/ESP, Triggerbot
- Generic: Noclip, inventory editing, Godmode, spambot

Maphack (SC2)


Achievements (F9)
Menu (F10)
Message Log (F11)
Help (F12)
Production (D)
7/11
7/11
ValiantChaos Maphack v1.2.9
Cracked by rolle3k a.k.a. Sheppard
0:03
Command Center
Training
1500/1500
2
3
4
5

Entity VAC (MapleStory)

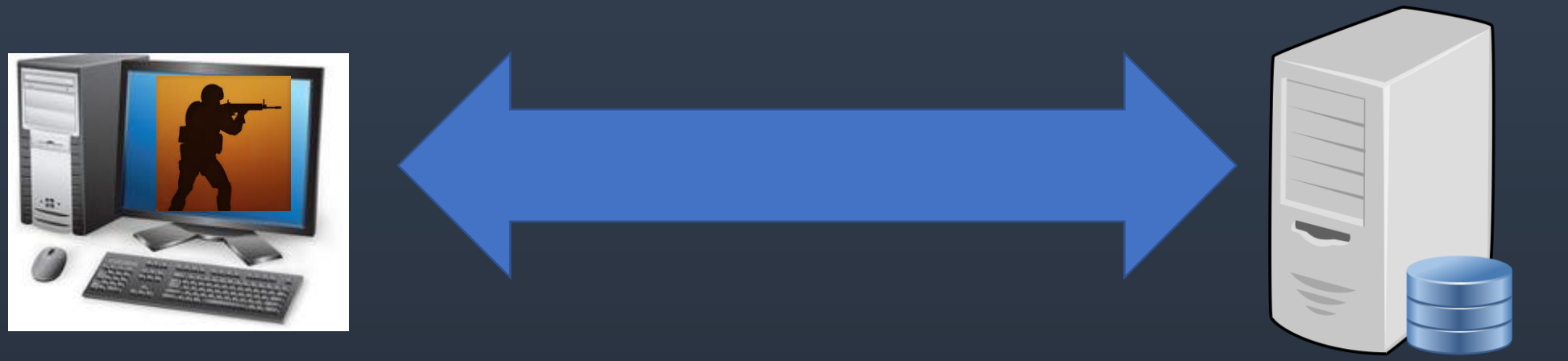
ESP/Chams (CSGO)

Multihack (Minecraft)

What is Possible in CS:GO?

We need to consider the security model of the game.

CS:GO's security model



Client

Server

CS:GO's security model

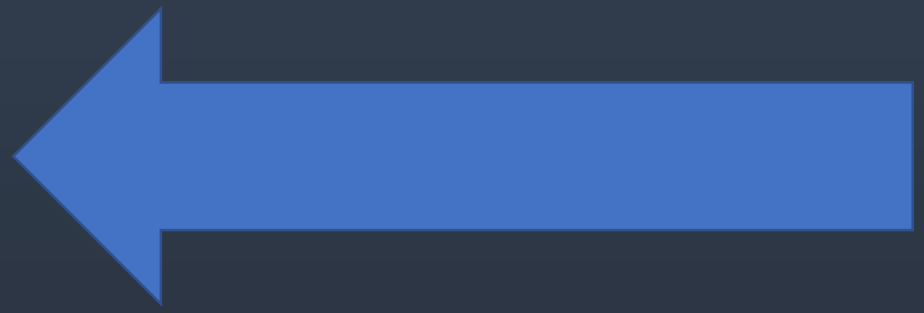
Client

- Input
- Rendering
- Physics

Server

CS:GO's security model

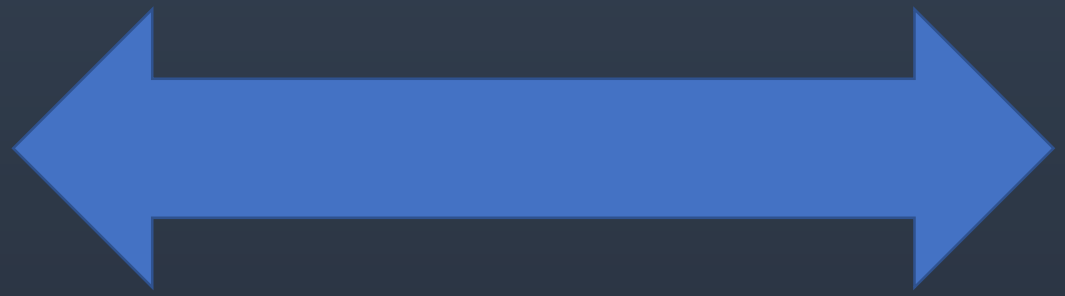
Client



Server

- Hit detection
- Entity HP
- Physics
- Networking

CS:GO's security model



Client

- Input
- Rendering
- Physics

Server

- Hit detection
- Entity HP
- Physics
- Networking

Let’s think about it!

Godmode

Is it possible?

Demo

Why didn't it work?

Entity HP is server side

CS:GO's security model

Client

Server

- Hit detection
- Entity HP
- Physics
- Networking

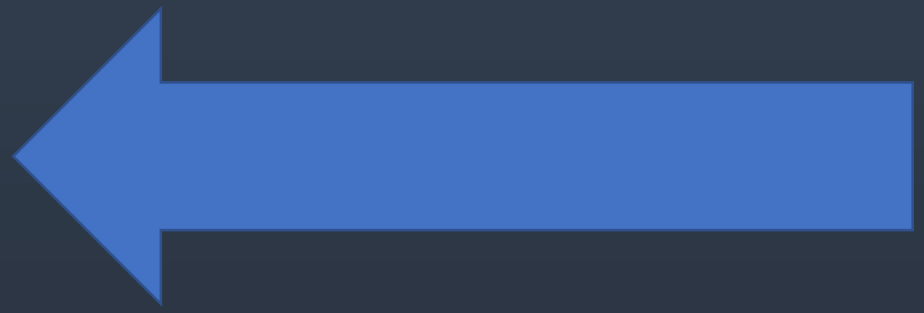
CS:GO's security model

Client

Server

- Hit detection
- Entity HP
- Physics
- Networking

CS:GO's security model



Client

-Entity HP

Server

- Hit detection
- Entity HP
- Physics
- Networking

Clientside vs. Serverside

- Input data (mouse, keyboard, etc.)
  - Angles
  - Shooting/attacking
  - Jumping/strafing/etc.
- Entity data*
- Weapon recoil*
- Map data/physics*

- Entity data (health, money, position)
- Score
- Map data/physics
- Hit detection
- Weapon recoil/spread

*read-only

The Big Idea

We can't "break" the game with cheats like godmode or noclip.

The Big Idea

Instead we make a program to play “perfectly”.

How to Play Perfectly

- Aim perfectly (Aimbot)
- Shoot perfectly (Triggerbot)
- Perfect knowledge (Wallhack/ESP)

We need to learn more.

First need to understand the game before we can interact with it.

Cheat Engine is memory hacking

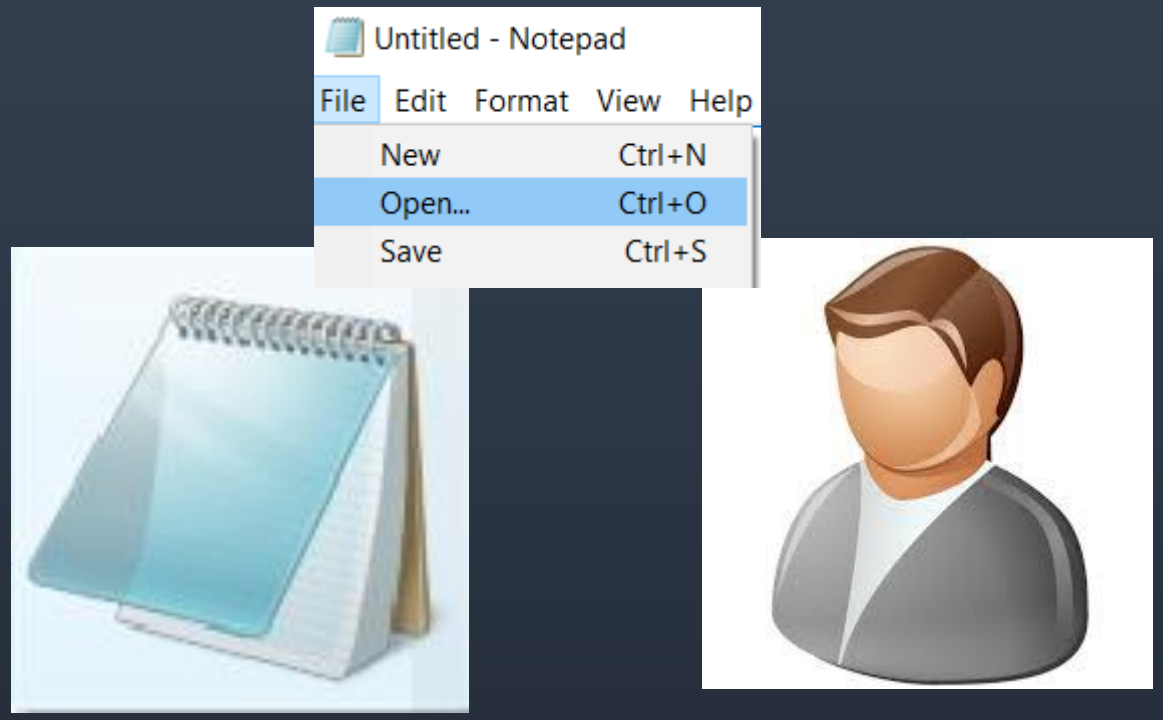
What exactly <u>is</u> memory?

Memory

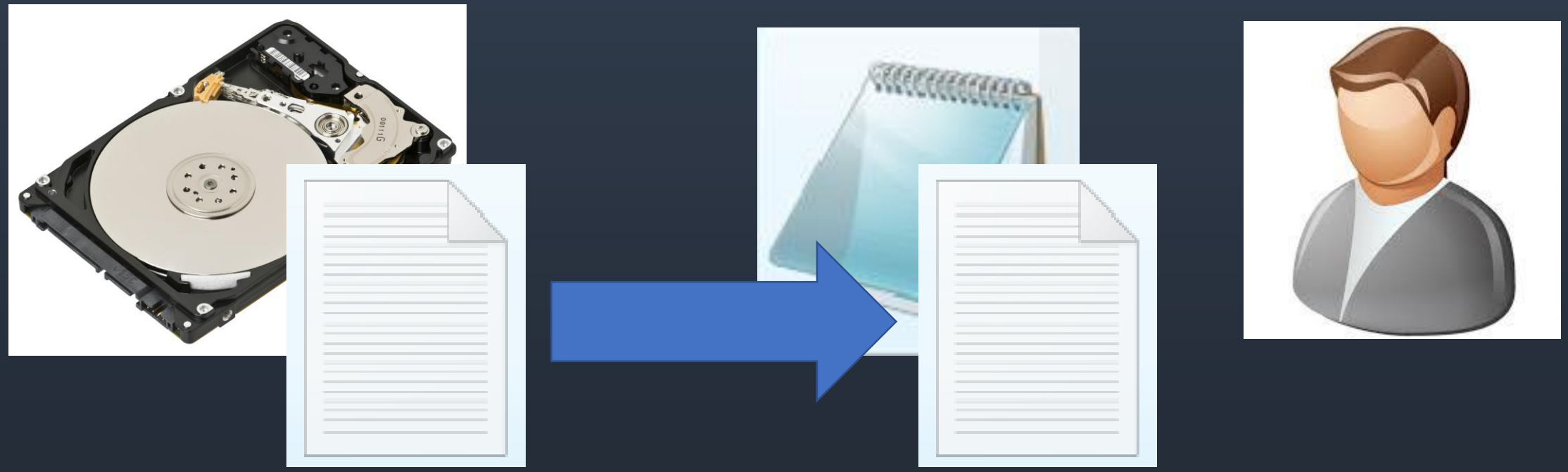
- Programs have two places to store data: on disk, and in memory
- Disk is slow, memory is fast
- Any data currently in use is in memory

An Example

An Example

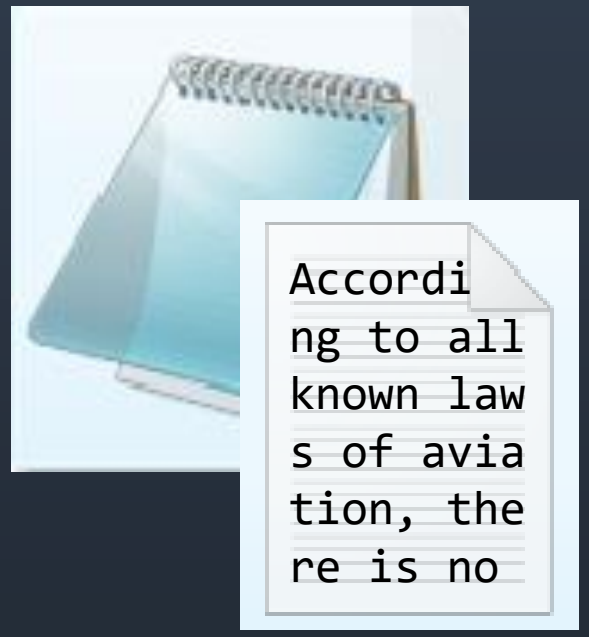

Untitled - Notepad
File
Edit
Format
View
Help
New
Ctrl+N
Open...
Ctrl+O
Save
Ctrl+S

An Example

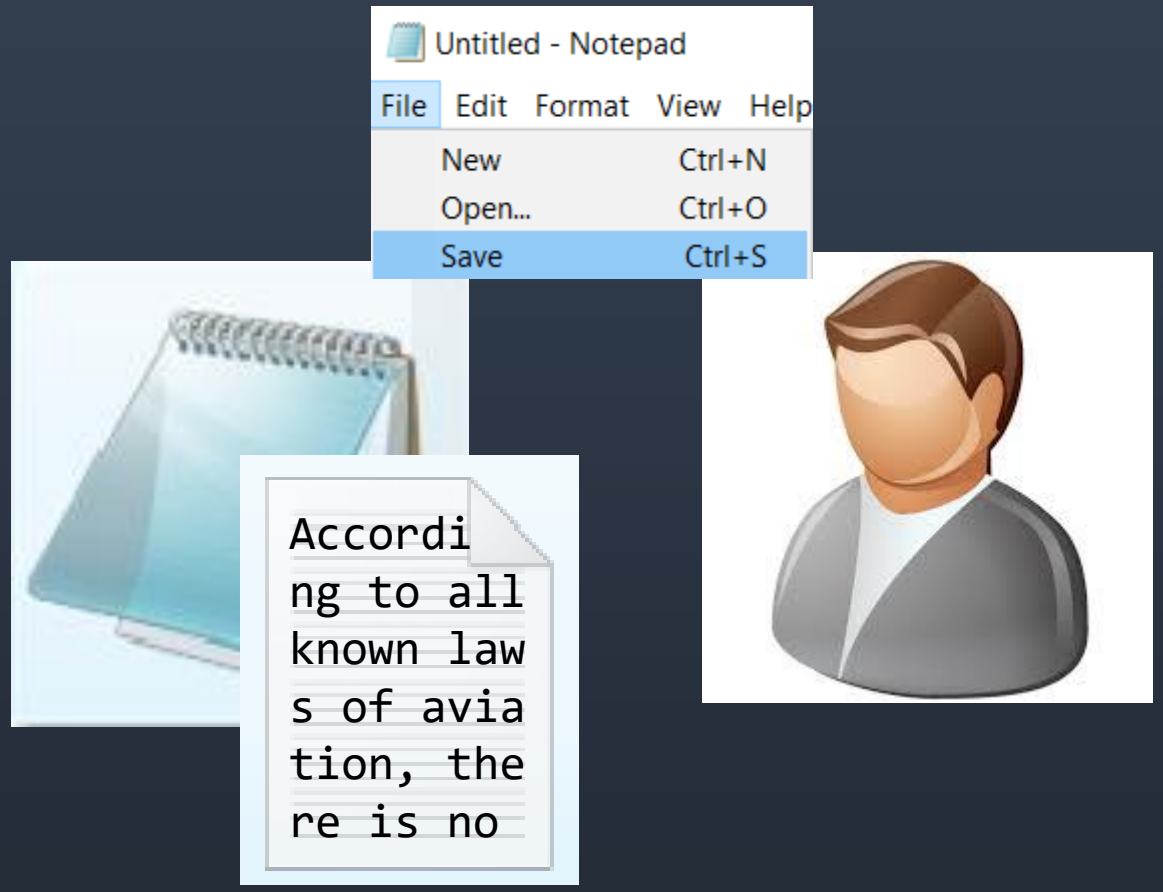


An Example

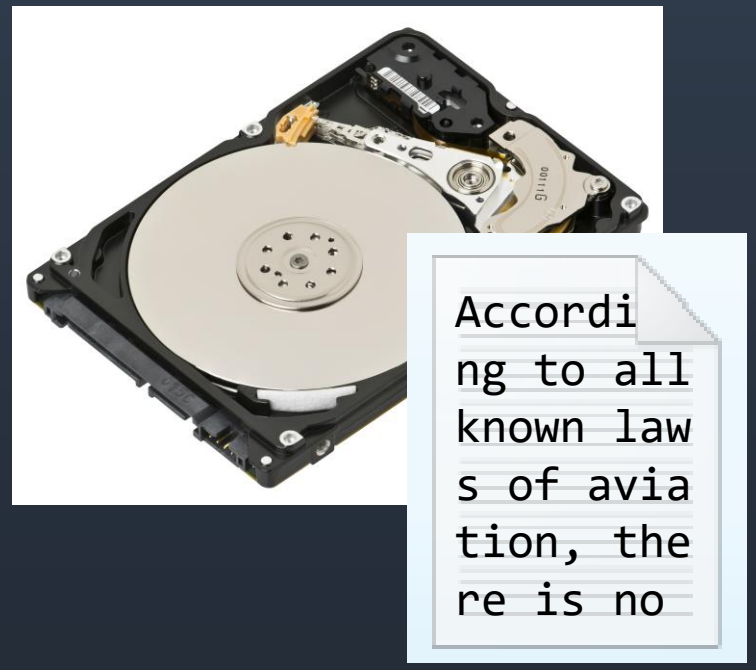
An Example

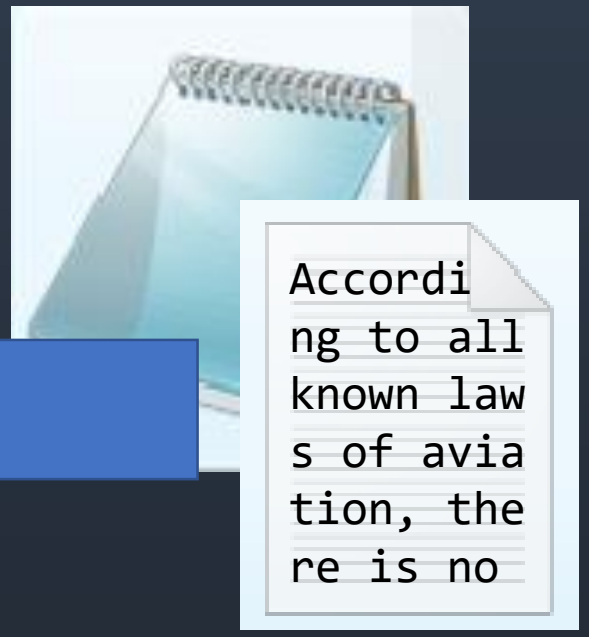


An Example

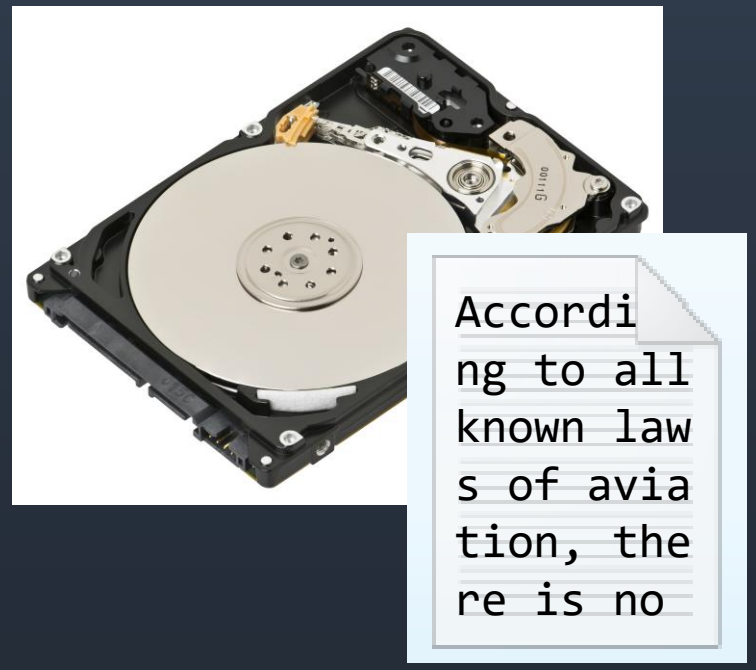

Untitled - Notepad
File Edit Format View Help
New Ctrl+N
Open... Ctrl+O
Save Ctrl+S
According to all known laws of aviation, there is no

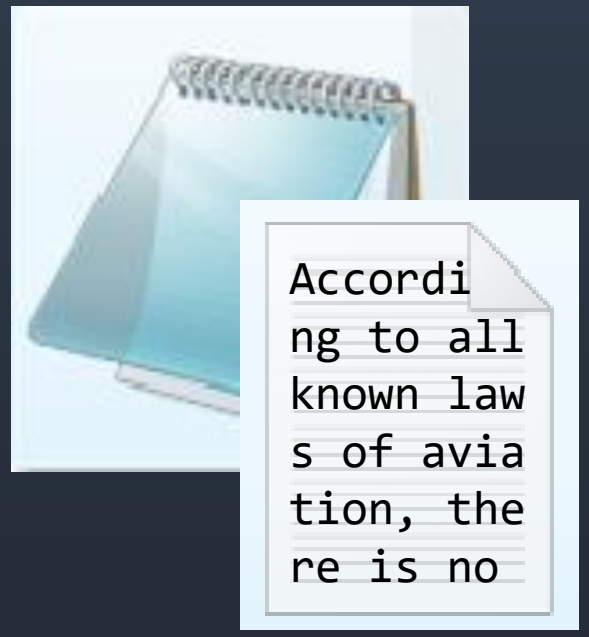
An Example


Accordi
ng to all
known law
s of avia
tion, the
re is no

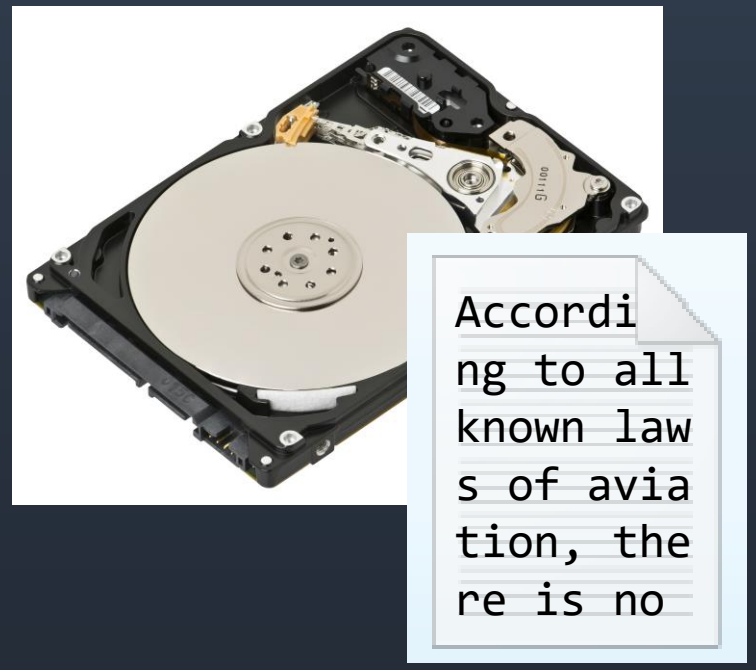

Accordi
ng to all
known law
s of avia
tion, the
re is no

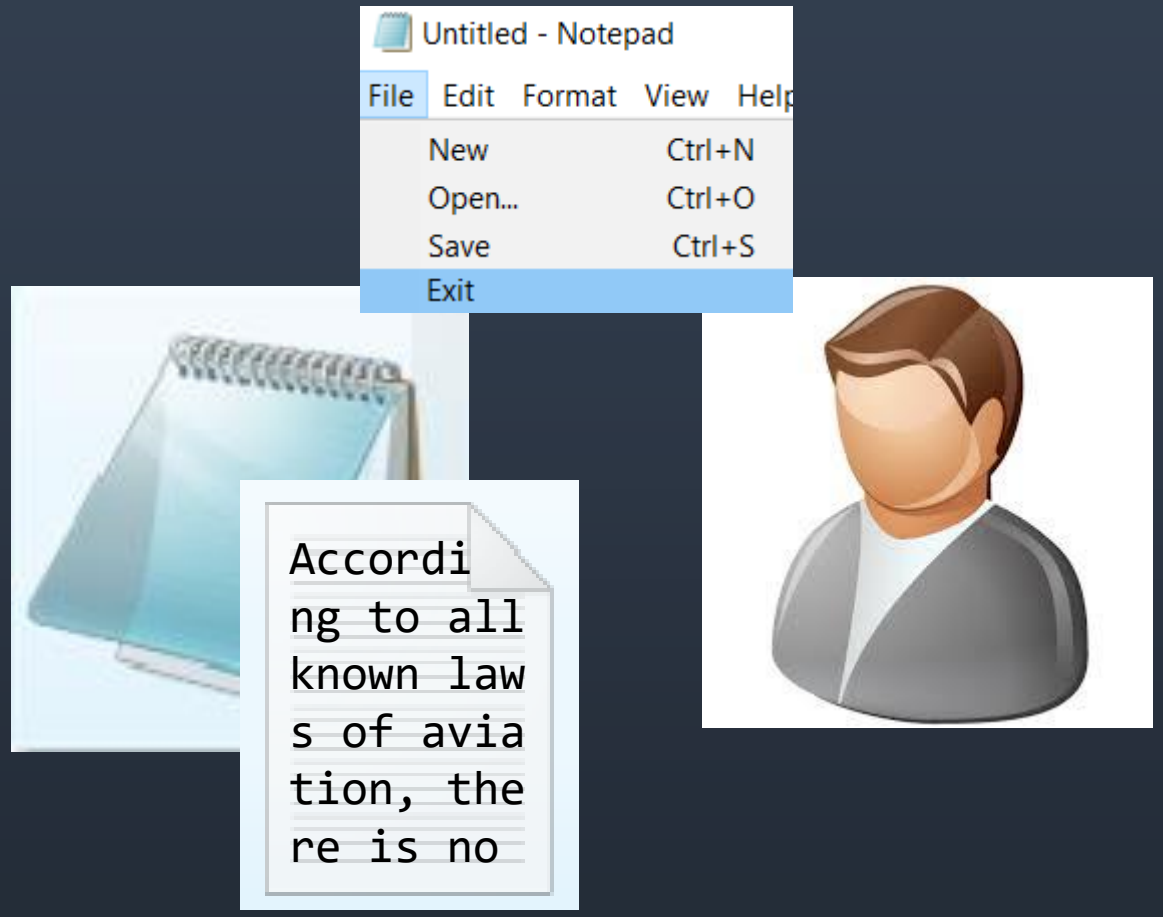
An Example


Accordi
ng to all
known law
s of avia
tion, the
re is no

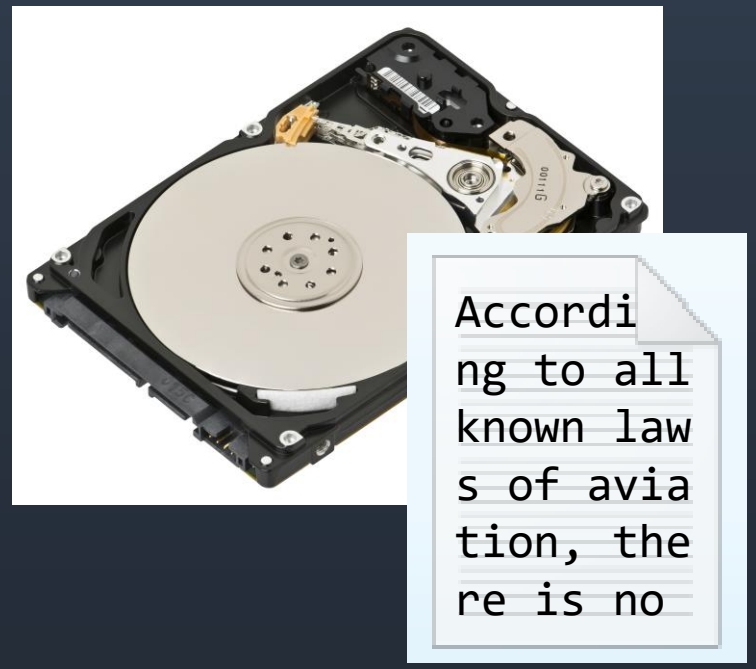

Accordi
ng to all
known law
s of avia
tion, the
re is no

An Example


Accordi
ng to all
known law
s of avia
tion, the
re is no


Untitled - Notepad
File Edit Format View Help
New Ctrl+N
Open... Ctrl+O
Save Ctrl+S
Exit
Accordi
ng to all
known law
s of avia
tion, the
re is no

An Example



What memory looks like (Demo)

```
Protect:Read/Write  AllocationBase=19317B10000 Base=19317B79000 Size=1E000
address     D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF 0123456789ABCDEF
19317B790D0 41 00 63 00 63 00 6F 00 72 00 64 00 69 00 6E 00 A.c.c.o.r.d.i.n.
19317B790E0 67 00 20 00 74 00 6F 00 20 00 61 00 6C 00 6C 00 g. .t.o. .a.l.l.
19317B790F0 20 00 6B 00 6E 00 6F 00 77 00 6E 00 20 00 6C 00  .k.n.o.w.n. .l.
19317B79100 61 00 77 00 73 00 20 00 6F 00 66 00 20 00 61 00 a.w.s. .o.f. .a.
19317B79110 76 00 69 00 61 00 74 00 69 00 6F 00 6E 00 2C 00 v.i.a.t.i.o.n.,.
19317B79120 20 00 74 00 68 00 65 00 72 00 65 00 20 00 69 00  .t.h.e.r.e. .i.
19317B79130 73 00 20 00 6E 00 6F 00 20 00 77 00 61 00 79 00 s. .n.o. .w.a.y.
19317B79140 20 00 61 00 20 00 62 00 65 00 65 00 20 00 73 00  .a. .b.e.e. .s.
19317B79150 68 00 6F 00 75 00 6C 00 64 00 20 00 62 00 65 00 h.o.u.l.d. .b.e.
19317B79160 20 00 61 00 62 00 6C 00 65 00 20 00 74 00 6F 00  .a.b.l.e. .t.o.
19317B79170 20 00 66 00 6C 00 79 00 2E 00 20 00 49 00 74 00  .f.l.y... .I.t.
19317B79180 73 00 20 00 77 00 69 00 6E 00 67 00 73 00 20 00 s. .w.i.n.g.s. .
19317B79190 61 00 72 00 65 00 20 00 74 00 6F 00 6F 00 20 00 a.r.e. .t.o.o. .
19317B791A0 73 00 6D 00 61 00 6C 00 6C 00 20 00 74 00 6F 00 s.m.a.l.l. .t.o.
19317B791B0 20 00 67 00 65 00 74 00 20 00 69 00 74 00 73 00  .g.e.t. .i.t.s.
19317B791C0 20 00 66 00 61 00 74 00 20 00 6C 00 69 00 74 00  .f.a.t. .l.i.t.
19317B791D0 74 00 6C 00 65 00 20 00 62 00 6F 00 64 00 79 00 t.l.e. .b.o.d.y.
19317B791E0 20 00 6F 00 66 00 66 00 20 00 74 00 68 00 65 00  .o.f.f. .t.h.e.
19317B791F0 20 00 67 00 72 00 6F 00 75 00 6E 00 64 00 2E 00  .g.r.o.u.n.d...
19317B79200 20 00 54 00 68 00 65 00 20 00 62 00 65 00 65 00  .T.h.e. .b.e.e.
```

What memory looks like (Demo)

```
Protect:Read/Write  AllocationBase=19317B10000 Base=19317B79000 Size=1E000
address     D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF 0123456789ABCDEF
19317B790D0 41 00 63 00 63 00 6F 00 72 00 64 00 69 00 6E 00 A.c.c.o.r.d.i.n.
19317B790E0 67 00 20 00 74 00 6F 00 20 00 61 00 6C 00 6C 00 g. .t.o. .a.l.l.
19317B790F0 20 00 6B 00 6E 00 6F 00 77 00 6E 00 20 00 6C 00  .k.n.o.w.n. .l.
19317B79100 61 00 77 00 73 00 20 00 6F 00 66 00 20 00 61 00 a.w.s. .o.f. .a.
19317B79110 76 00 69 00 61 00 74 00 69 00 6F 00 6E 00 2C 00 v.i.a.t.i.o.n.,.
19317B79120 20 00 74 00 68 00 65 00 72 00 65 00 20 00 69 00  .t.h.e.r.e. .i.
19317B79130 73 00 20 00 6E 00 6F 00 20 00 77 00 61 00 79 00 s. .n.o. .w.a.y.
19317B79140 20 00 61 00 20 00 62 00 65 00 65 00 20 00 73 00  .a. .b.e.e. .s.
19317B79150 68 00 6F 00 75 00 6C 00 64 00 20 00 62 00 65 00 h.o.u.l.d. .b.e.
19317B79160 20 00 61 00 62 00 6C 00 65 00 20 00 74 00 6F 00  .a.b.l.e. .t.o.
19317B79170 20 00 66 00 6C 00 79 00 2E 00 20 00 49 00 74 00  .f.l.y... .I.t.
19317B79180 73 00 20 00 77 00 69 00 6E 00 67 00 73 00 20 00 s. .w.i.n.g.s. .
19317B79190 61 00 72 00 65 00 20 00 74 00 6F 00 6F 00 20 00 a.r.e. .t.o.o. .
19317B791A0 73 00 6D 00 61 00 6C 00 6C 00 20 00 74 00 6F 00 s.m.a.l.l. .t.o.
19317B791B0 20 00 67 00 65 00 74 00 20 00 69 00 74 00 73 00  .g.e.t. .i.t.s.
19317B791C0 20 00 66 00 61 00 74 00 20 00 6C 00 69 00 74 00  .f.a.t. .l.i.t.
19317B791D0 74 00 6C 00 65 00 20 00 62 00 6F 00 64 00 79 00 t.l.e. .b.o.d.y.
19317B791E0 20 00 6F 00 66 00 66 00 20 00 74 00 68 00 65 00  .o.f.f. .t.h.e.
19317B791F0 20 00 67 00 72 00 6F 00 75 00 6E 00 64 00 2E 00  .g.r.o.u.n.d...
19317B79200 20 00 54 00 68 00 65 00 20 00 62 00 65 00 65 00  .T.h.e. .b.e.e.
```

Memory is made of bytes

- Memory is a linear tape
- Each cell holds 1 byte
- 1 byte is 8 bits (0 to 255)
- Each cell has an address

Addresses are key!

Addresses tell us where in memory the values we care about are.

Addresses are key!

We need to know the address of a value to read or write to it.

Addresses are key!

Our hack is useless if we don't know any addresses.

What memory looks like (Demo)

```
Protect:Read/Write  AllocationBase=19317B10000 Base=19317B79000 Size=1E000
address     D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF 0123456789ABCDEF
19317B790D0 41 00 63 00 63 00 6F 00 72 00 64 00 69 00 6E 00 A.c.c.o.r.d.i.n.
19317B790E0 67 00 20 00 74 00 6F 00 20 00 61 00 6C 00 6C 00 g. .t.o. .a.l.l.
19317B790F0 20 00 6B 00 6E 00 6F 00 77 00 6E 00 20 00 6C 00  .k.n.o.w.n. .l.
19317B79100 61 00 77 00 73 00 20 00 6F 00 66 00 20 00 61 00 a.w.s. .o.f. .a.
19317B79110 76 00 69 00 61 00 74 00 69 00 6F 00 6E 00 2C 00 v.i.a.t.i.o.n.,.
19317B79120 20 00 74 00 68 00 65 00 72 00 65 00 20 00 69 00  .t.h.e.r.e. .i.
19317B79130 73 00 20 00 6E 00 6F 00 20 00 77 00 61 00 79 00 s. .n.o. .w.a.y.
19317B79140 20 00 61 00 20 00 62 00 65 00 65 00 20 00 73 00  .a. .b.e.e. .s.
19317B79150 68 00 6F 00 75 00 6C 00 64 00 20 00 62 00 65 00 h.o.u.l.d. .b.e.
19317B79160 20 00 61 00 62 00 6C 00 65 00 20 00 74 00 6F 00  .a.b.l.e. .t.o.
19317B79170 20 00 66 00 6C 00 79 00 2E 00 20 00 49 00 74 00  .f.l.y... .I.t.
19317B79180 73 00 20 00 77 00 69 00 6E 00 67 00 73 00 20 00 s. .w.i.n.g.s. .
19317B79190 61 00 72 00 65 00 20 00 74 00 6F 00 6F 00 20 00 a.r.e. .t.o.o. .
19317B791A0 73 00 6D 00 61 00 6C 00 6C 00 20 00 74 00 6F 00 s.m.a.l.l. .t.o.
19317B791B0 20 00 67 00 65 00 74 00 20 00 69 00 74 00 73 00  .g.e.t. .i.t.s.
19317B791C0 20 00 66 00 61 00 74 00 20 00 6C 00 69 00 74 00  .f.a.t. .l.i.t.
19317B791D0 74 00 6C 00 65 00 20 00 62 00 6F 00 64 00 79 00 t.l.e. .b.o.d.y.
19317B791E0 20 00 6F 00 66 00 66 00 20 00 74 00 68 00 65 00  .o.f.f. .t.h.e.
19317B791F0 20 00 67 00 72 00 6F 00 75 00 6E 00 64 00 2E 00  .g.r.o.u.n.d...
19317B79200 20 00 54 00 68 00 65 00 20 00 62 00 65 00 65 00  .T.h.e. .b.e.e.
```

Do Addresses Change? (Demo)

Static vs dynamic addresses

- Some addresses are always the same (across restarts, different game servers, etc): static
- Most addresses change: dynamic
- Static addresses are relative to a "module base"

What’s a module? (Demo)

Modules

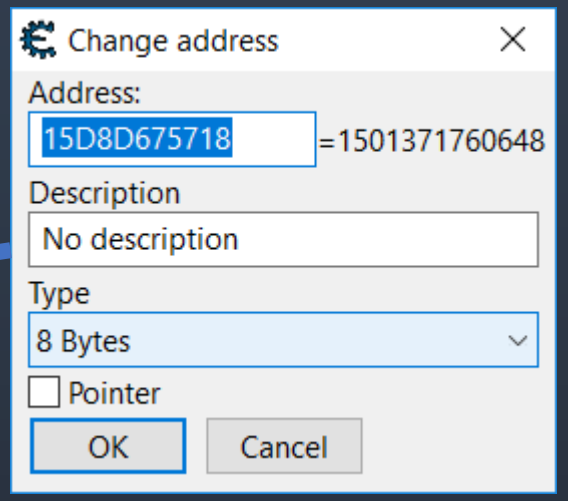
- Modules are basically any file that is mapped (copied) into the program's address space.
- The program itself is the "main module"

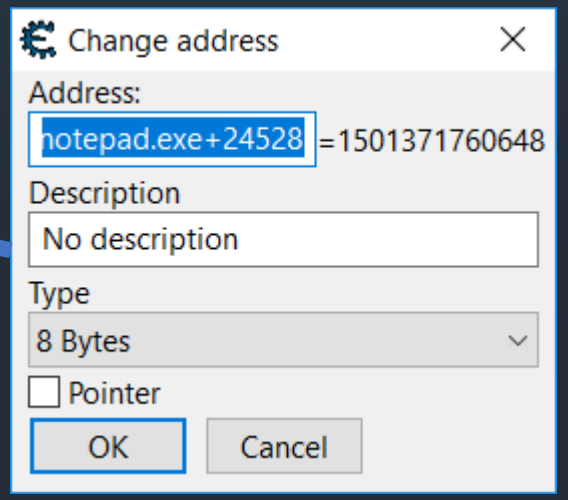
Examples of modules

- csgo.exe
- Kernel32.dll
- ntdll.dll
- client_panorama.dll
- notepad.exe

“Relative to a module base”

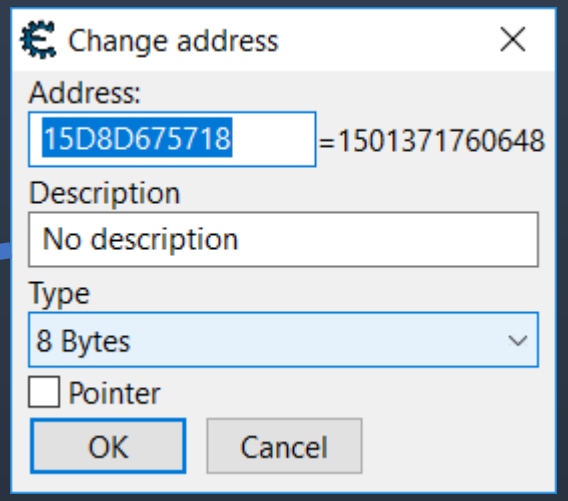
Address	Value	Previous
15D8D604648	15D90BB0008	15D90BB0008
15D8D63F630	15D90BB0008	15D90BB0008
15D8D675718	15D90BB0008	15D90BB0008
7FF682274528	15D90BB0008	15D90BB0008

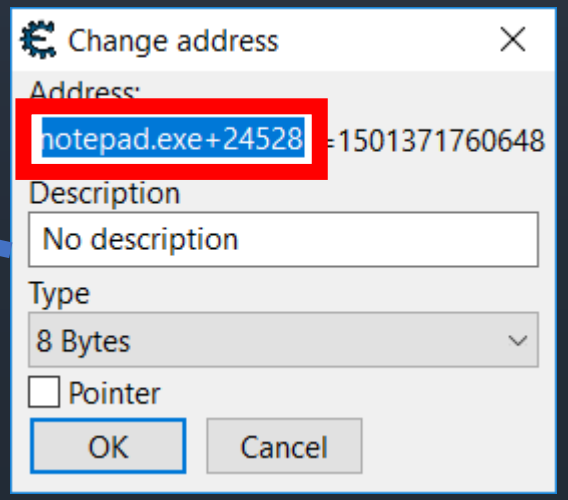




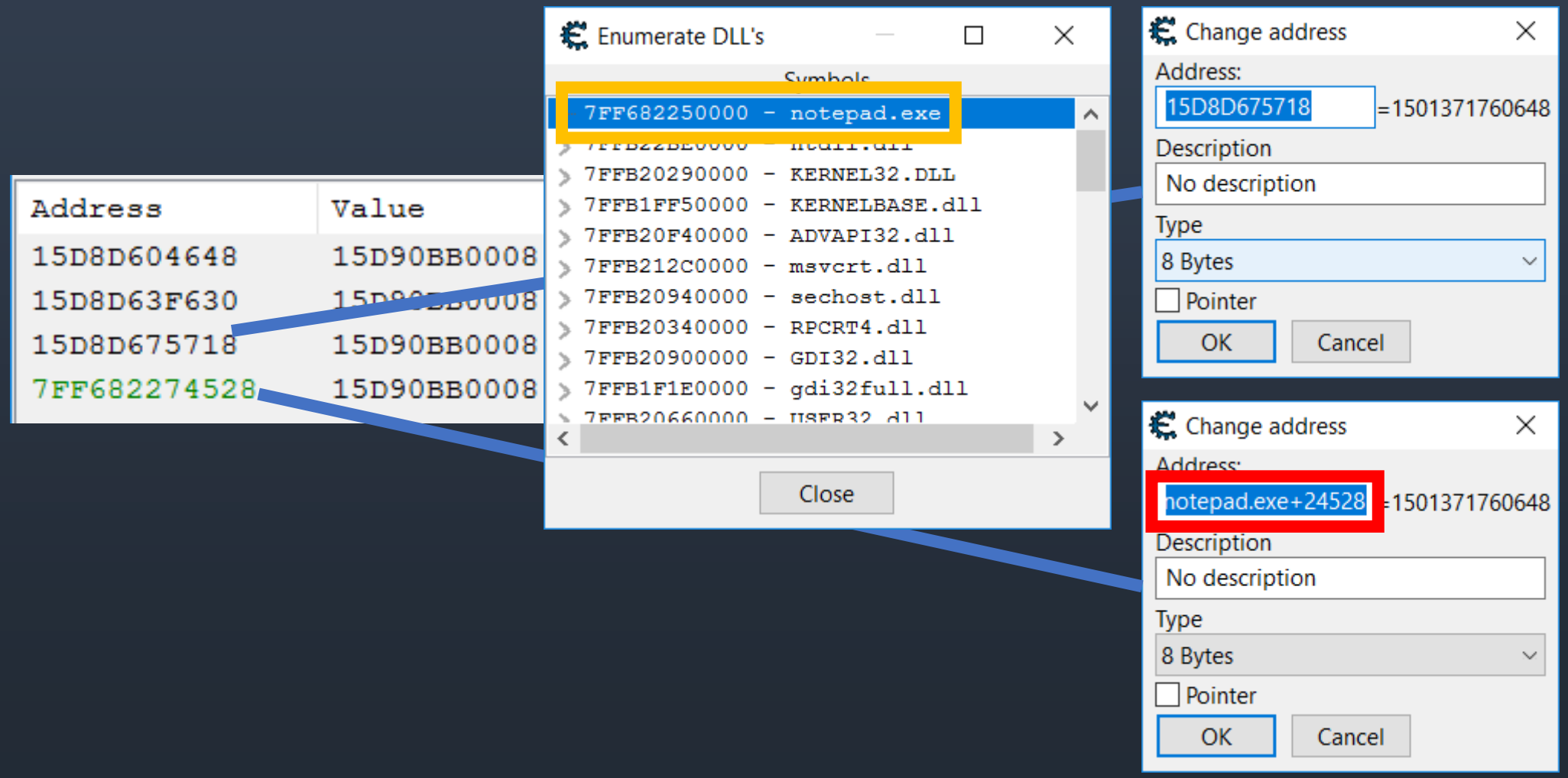
“Relative to a module base”

Address	Value	Previous
15D8D604648	15D90BB0008	15D90BB0008
15D8D63F630	15D90BB0008	15D90BB0008
15D8D675718	15D90BB0008	15D90BB0008
7FF682274528	15D90BB0008	15D90BB0008

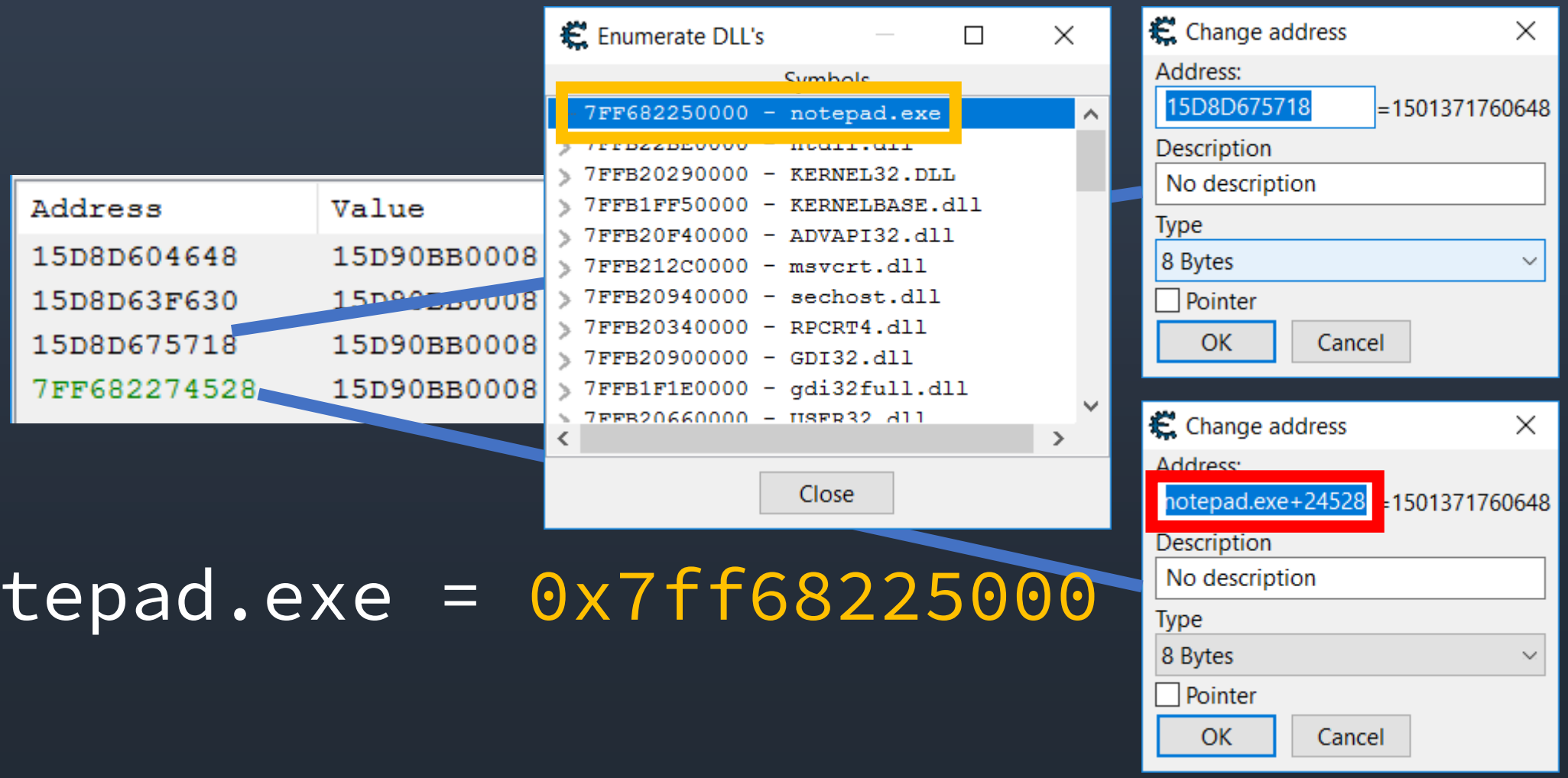




“Relative to a module base”

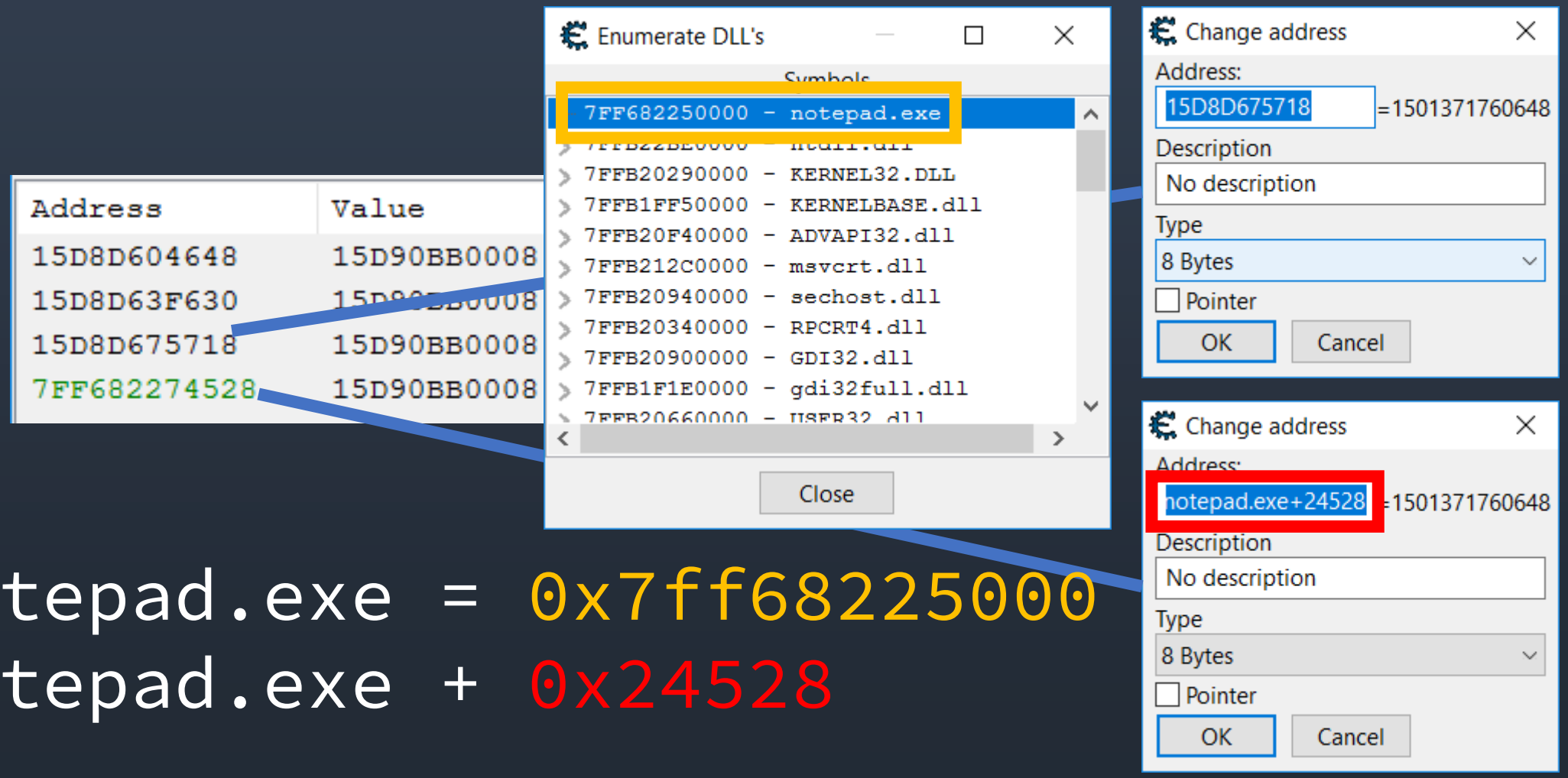


“Relative to a module base”



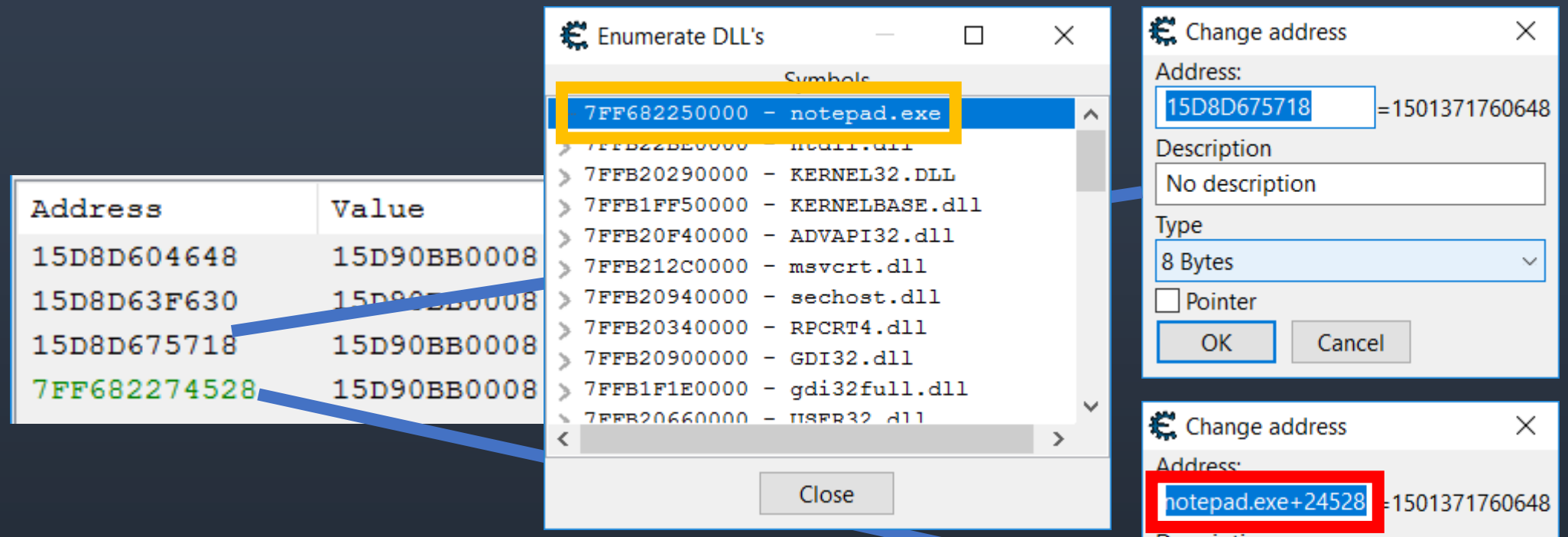
notepad.exe = 0x7ff68225000

“Relative to a module base”



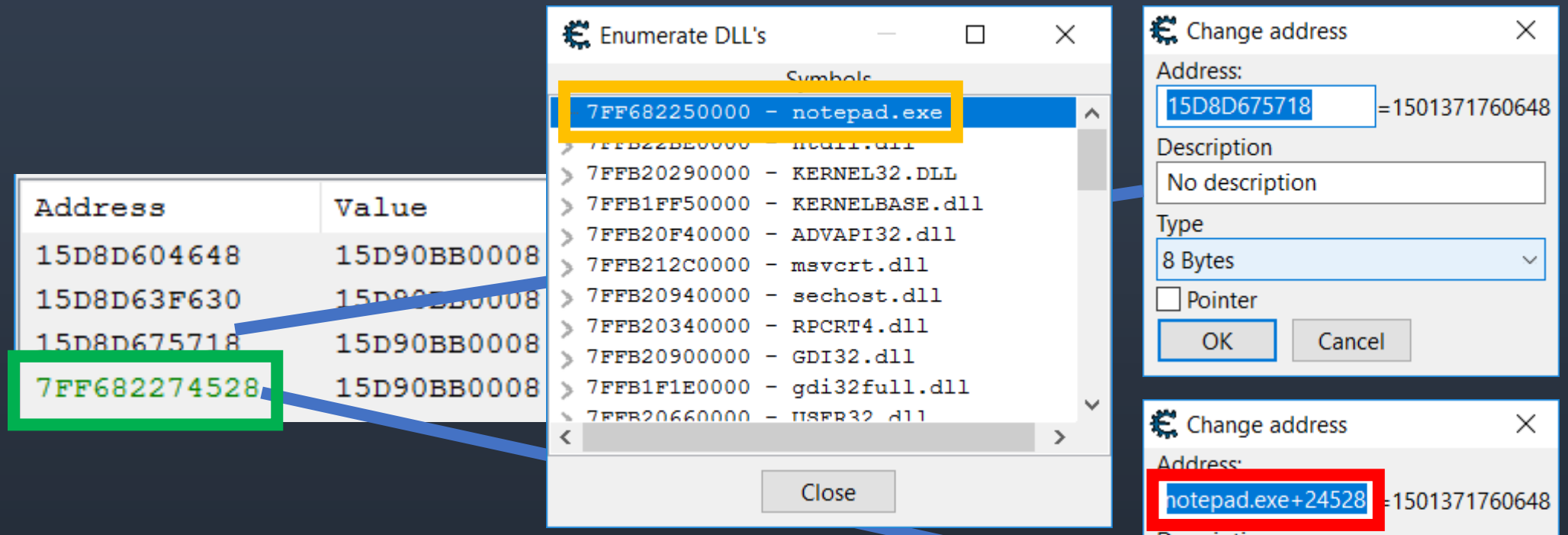

```
notepad.exe = 0x7ff68225000
notepad.exe + 0x24528
```

“Relative to a module base”




```
notepad.exe = 0x7ff68225000
notepad.exe + 0x24528
            = 0x7ff68249528
```

“Relative to a module base”




```
notepad.exe = 0x7ff68225000
notepad.exe + 0x24528
            = 0x7ff68249528
```

Even if **`notepad.exe`** restarts,
we can still find this value.

Why module base?

- We can easily find out the module base at any time.
- If we can base all of our addresses off of a module base, then we can deduce them at any time.
- That lets us know which part of memory to read/write.

Conclusion

- We hack by modifying the game's memory to modify its behavior.
- Client-side vs. server-side, godmode isn't possible.
- Some copies of values are dependent, i.e. not useful.
- We need to know the address of a value to read or write it.
- We can calculate addresses based off modules.

For next week...

- Reading: “Pointers for REAL Dummies”
- Take-home lab
- Check Slack

How to Setup CS:GO for hacking