RUSHING B ANYWAY, BLYAT!

BY @CAPTNBANANA

BSIDES MUNICH 2020

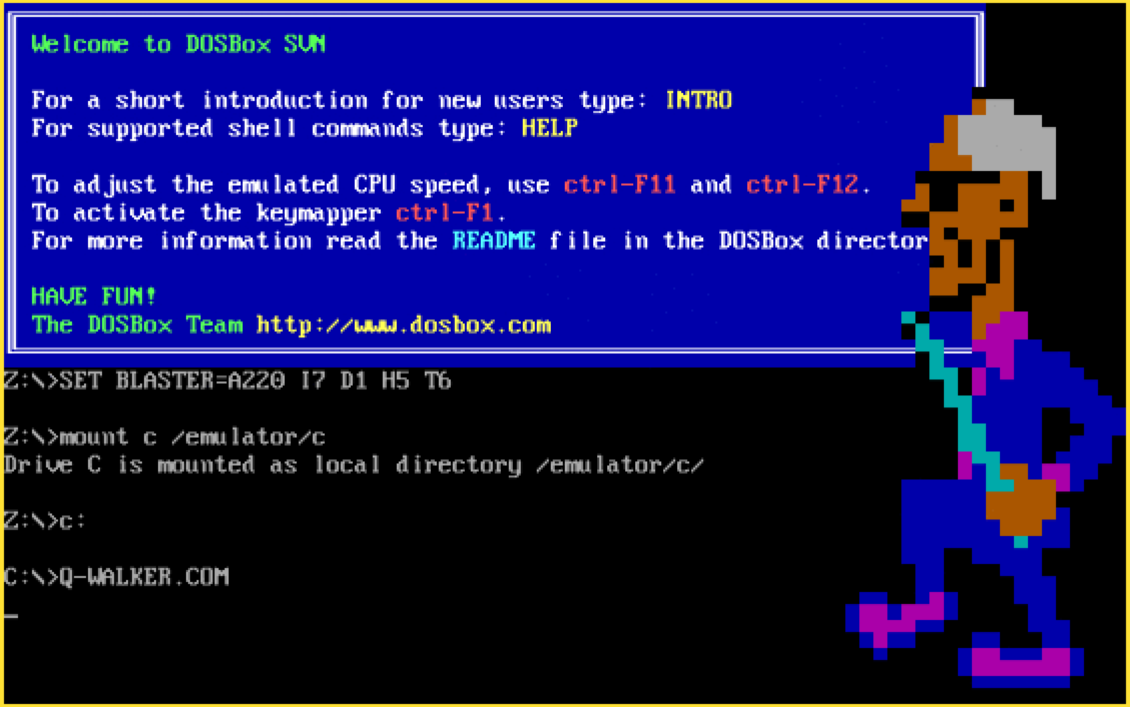
WHO R U MAN

- I do computer stuff!
- https://bananamafia.dev/tags/gamehacking/

MOTIVATION

MOTIVATION ($$$)

MOTIVATION



```
Welcome to DOSBox SVN

For a short introduction for new users type: INTRO
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox director

HAVE FUN!
The DOSBox Team http://www.dosbox.com
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c /emulator/c
Drive C is mounted as local directory /emulator/c/

Z:\>c:

C:\>Q-WALKER.COM
_
```

TOOLING

- Visual Studio
- Debugger, e.g. x64dbg
- RE tool of choice, e.g. radare2/Cutter/Ghidra
- **CheatEngine**
  - Windows and Linux (`ceserver`)
  - Run as admin/root (yolo)

TYPES OF GAME HACKS

- Internal
- External
- (Instrumented?)

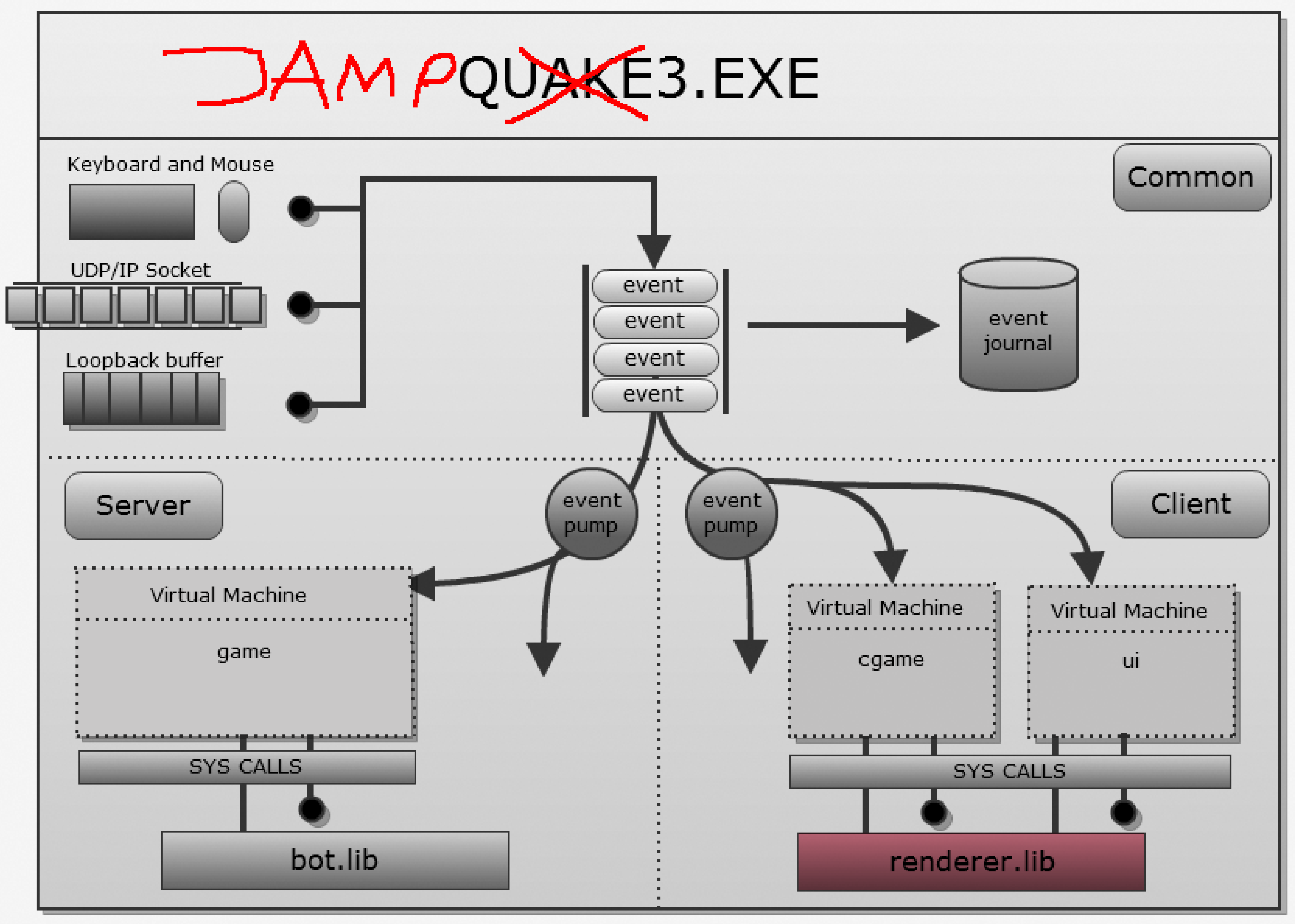
IDTECH3 ENGINE HACK

- "Quake3 Engine"
- For the game Star Wars: Jedi Knight - Jedi Academy
- For Windows

CHECKING OUT THE GAME ENGINE

- Hacks are engine-specific
- Understand what's implemented where
- Understand the rough program flow

THE IDTECH3 ENGINE



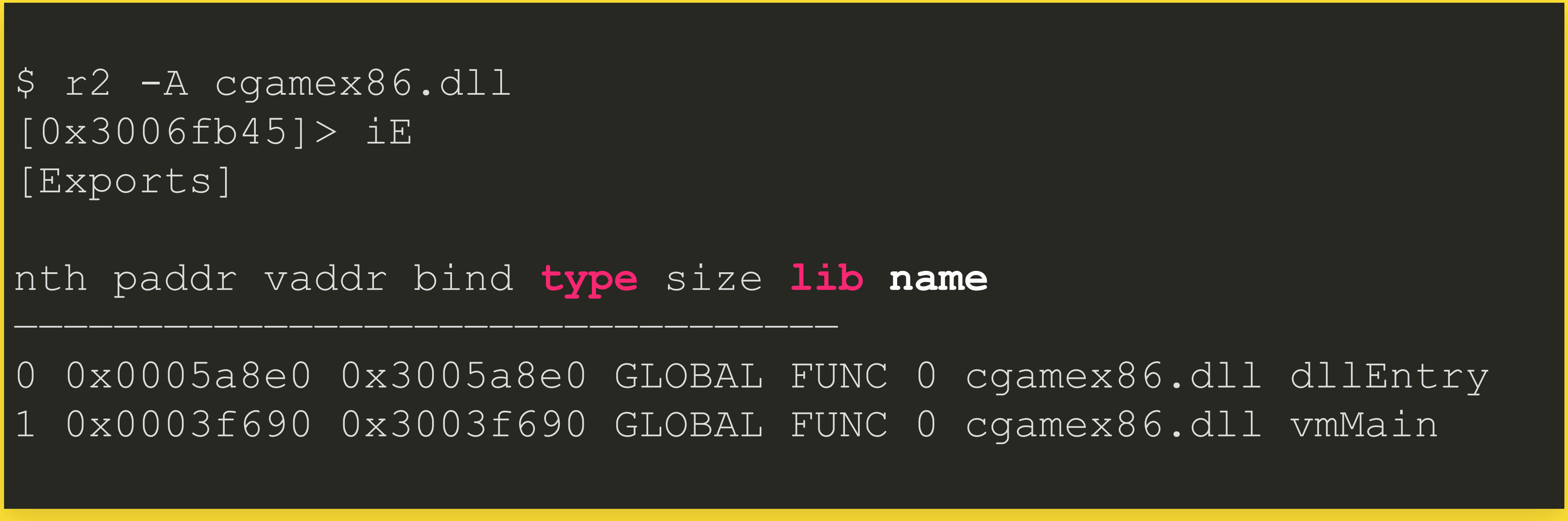
QVMS

- `cgame` QVM predicts local player states
- ^ good target to hook
- In-depth architecture analysis found here
- It's good

CGAME QVM

- Implemented in separate DLL: `cgamex86.dll`
- With exactly two exports

cgamex86.dll EXPORTS



```
$ r2 -A cgamex86.dll
[0x3006fb45]> iE
[Exports]

nth paddr vaddr bind type size lib name
_________________________________
0 0x0005a8e0 0x3005a8e0 GLOBAL FUNC 0 cgamex86.dll dllEntry
1 0x0003f690 0x3003f690 GLOBAL FUNC 0 cgamex86.dll vmMain
```

CGAMEX86.DLL EXPORTS: `vmMain()`

- Dispatcher from main executable (`jamp.exe`)
- Used for calls from `jamp.exe` -> `cgamex86.dll`
- **Hooked to execute own code (e.g. Aimbots)**
- Events: game load, frame drawn

```
vmMain(int command, int arg1, int arg2,int arg3,int arg4,int a
arg7,int arg8,int arg9,int arg10,int arg11,int arg12)
```

CGAMEX86.DLL EXPORTS: DLLENTRY()

- Callback from `cgame` QVM into `jamp.exe`
- Receives function pointer as parameter
- **Hooked to manipulate existing code (e.g. for Wallhack)**
- Events: Entity added, entity moves, game data received from server

```
Q_EXPORT void dllEntry(intptr_t (QDECL *syscallptr)( intptr_t
    Q_syscall = syscallptr;
    TranslateSyscalls();
}
```

HOW TO HOOK: EXAMPLE

1. `jamp.exe` wants to call `dllEntry()` of `cgamex86.dll`
2. `jamp.exe` loads `cgamex86.dll`
3. `jamp.exe` calls `GetProcAddress()` for `dllEntry()`
4. `jamp.exe` executes `dllEntry@Address`

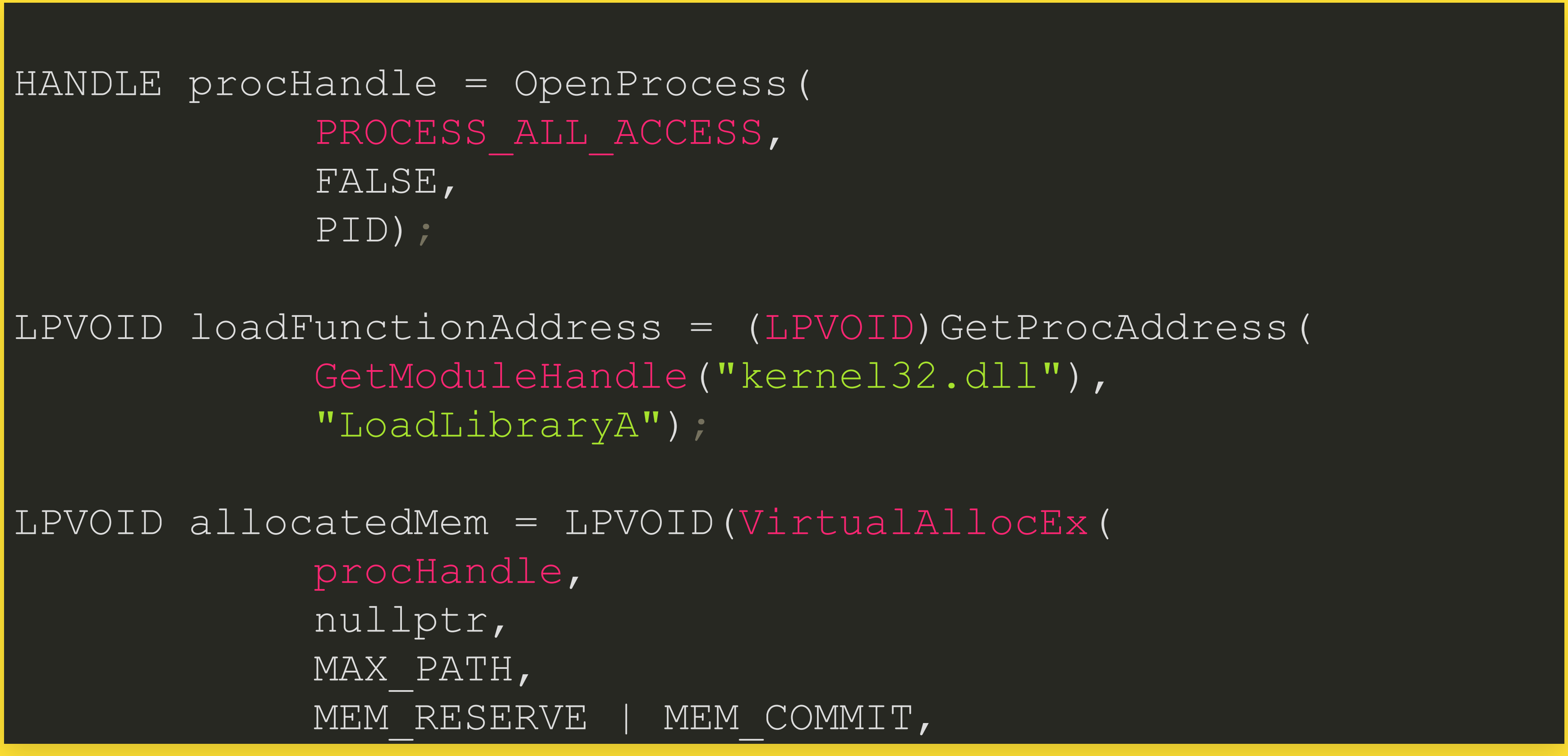
HOW TO HOOK: PLAN

- Hook `GetProcAddress()` for `jamp.exe`
- Replace returned function with own implementation
- Lastly call original function

DLL INJECTION

- Hack injects custom code into the game
- Easy method: DLL Injection
- Build loader and a DLL
- -> Internal hook based cheat

LOADER CODE



```
HANDLE procHandle = OpenProcess(
            PROCESS_ALL_ACCESS,
            FALSE,
            PID);

LPVOID loadFunctionAddress = (LPVOID)GetProcAddress(
            GetModuleHandle("kernel32.dll"),
            "LoadLibraryA");

LPVOID allocatedMem = LPVOID(VirtualAllocEx(
            procHandle,
            nullptr,
            MAX_PATH,
            MEM_RESERVE | MEM_COMMIT,
```

CREATING THE DLL

- After `CreateRemoteThread()`, `DllMain()` gets called
- Not that stealthy though

```
BOOL APIENTRY DllMain (HMODULE hModule, DWORD ul_reason_for_ca
    switch (ul_reason_for_call) {
        case DLL_PROCESS_ATTACH:
            MessageBox(0, "EYO ITS WORKING", "DLL", 0);
            break;
    }
    return TRUE;
}
```

HOOK SETUP

- Use hooking library, e.g. mhook

```
Mhook_SetHook(
    (PVOID*)&originalGetProcAddress,
    hookGetProcAddress
);
```

HOOK SETUP

- Redirect into own `dllEntry()`

```
if (isSubstr(lpProcName, "dllEntry")) {
    return (PROC)hookDLLEntry;
}
return (FARPROC)originalGetProcAddress(hModule, lpProcName);
```

HOOK SETUP

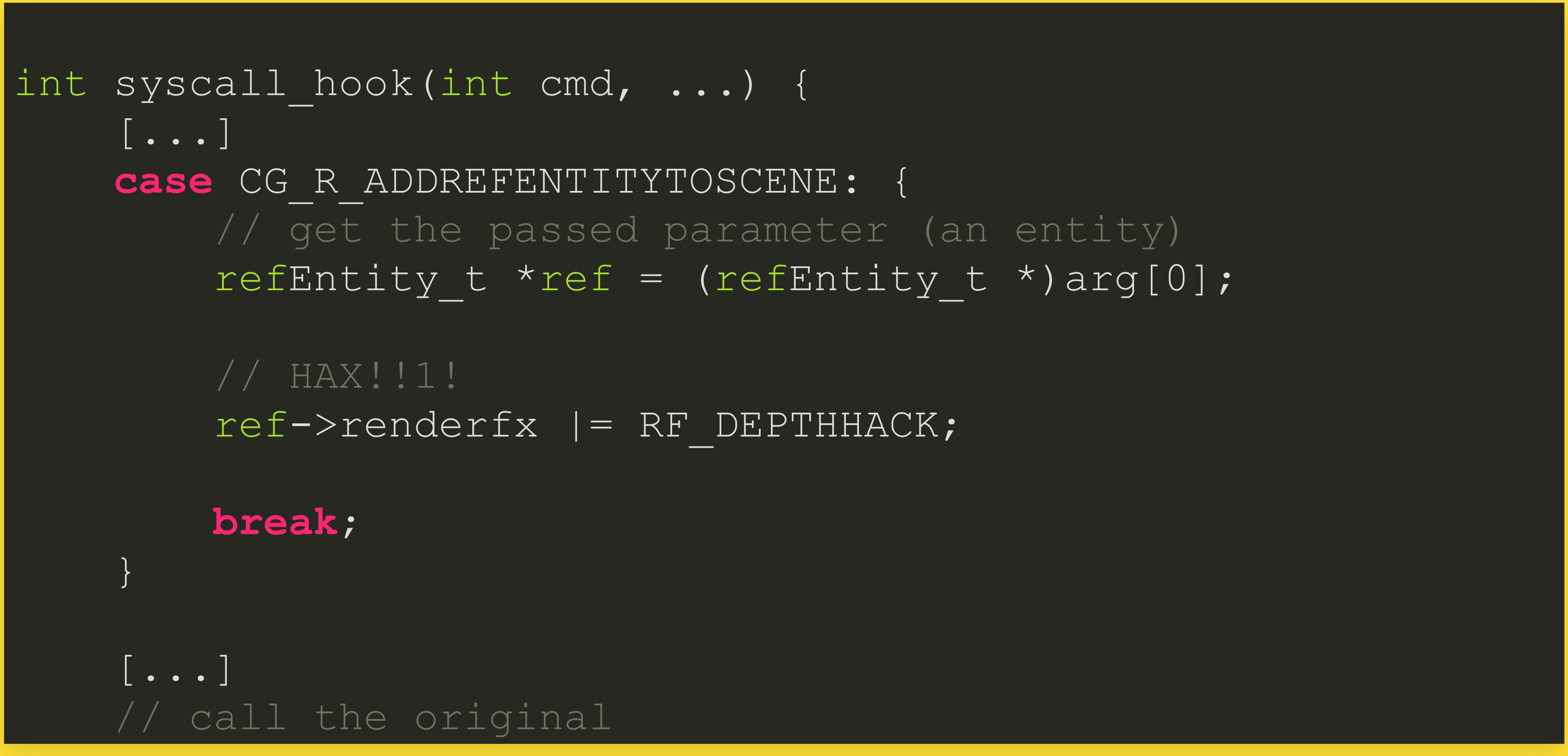
- Steal the parameter

```
void hookDLLEntry(int(QDECL *syscallptr)(int arg, ...)) {
    // steal original pointer
    syscall = syscallptr;
    // execute own function
    originalDLLEntry(syscall_hook);
}
```

THE ACTUAL HACK

- Goal: Wallhack
- Intercept function that adds entities, e.g. players
- Tip: Integrate released SDK

DEPTHHACK



```
int syscall_hook(int cmd, ...) {
    [...]
    case CG_R_ADDREFENTITYTOSCENE: {
        // get the passed parameter (an entity)
        refEntity_t *ref = (refEntity_t *)arg[0];

        // HAX!!1!
        ref->renderfx |= RF_DEPTHHACK;

        break;
    }

    [...]
    // call the original
```

DEMO

CS:GO AIMBOT

- Source Engine
- For Linux

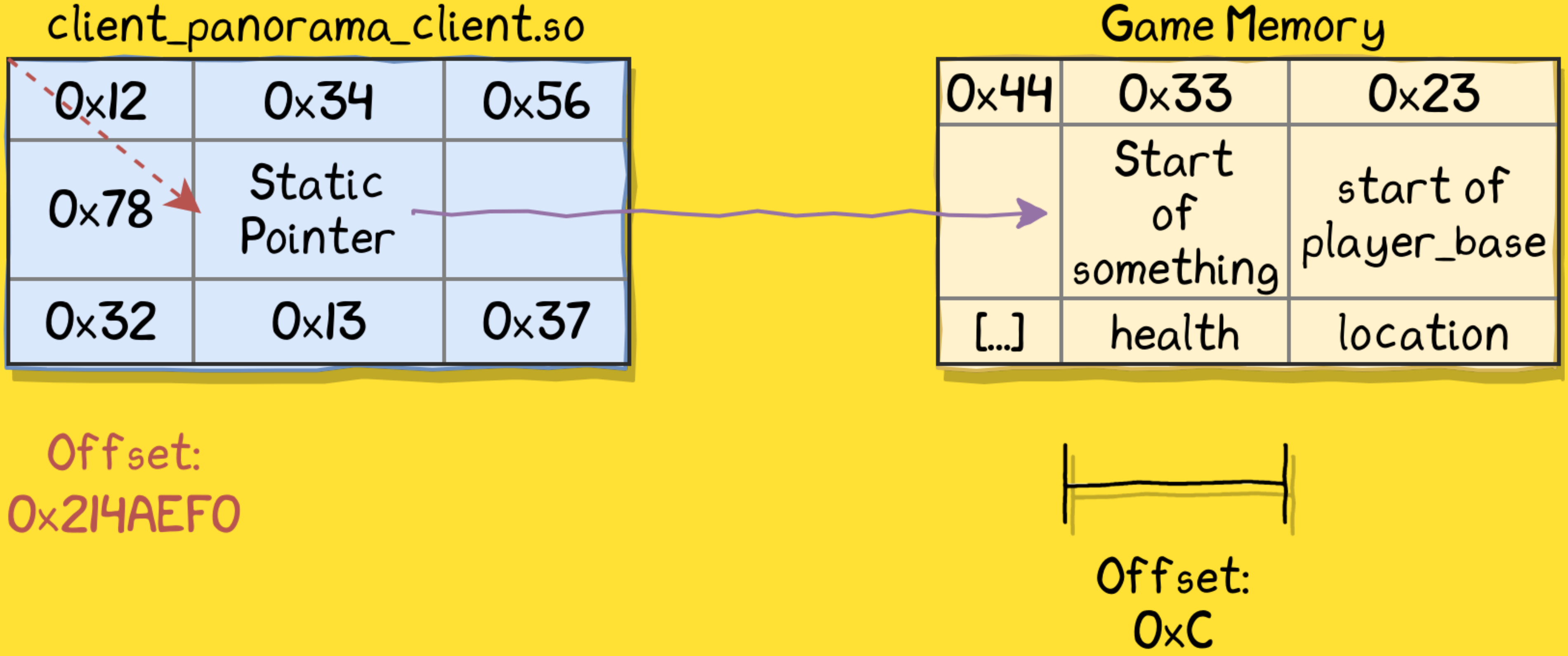
TOOLING

- CheatEngine
- `/proc/pid/maps`

HOW TO HACK

- Find own player struct in memory
- Find list of enemies in memory
- Get coordinates of enemies
- Get nearest enemy
- Adjust aim (using crazy math)

MEMORY ANALYSIS: STATIC POINTER



client_panorama_client.so

0x12	0x34	0x56
0x78	Static Pointer	
0x32	0x13	0x37

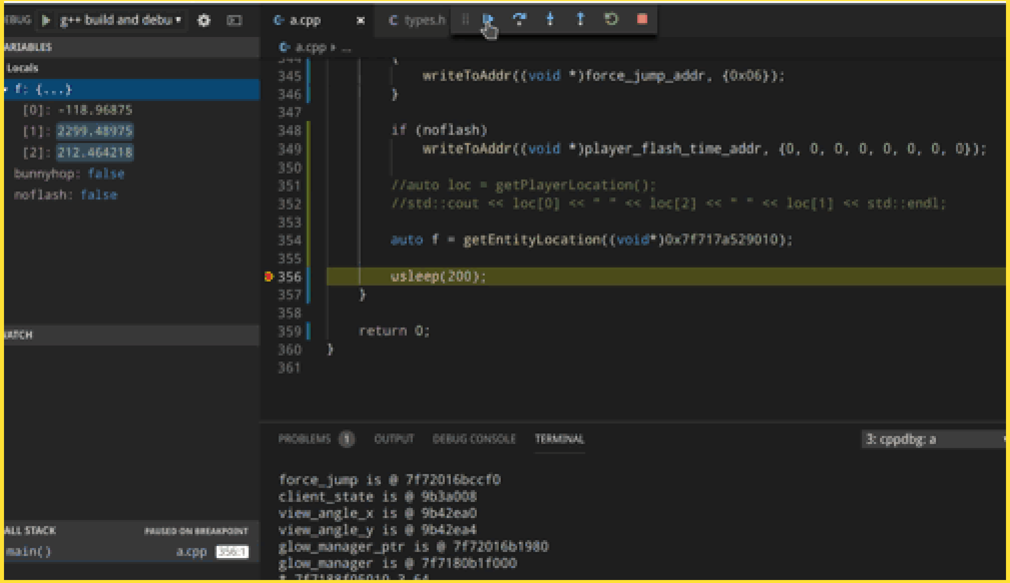
Offset: 0x214AEF0

Game Memory

0x44	0x33	0x23
	Start of something	start of player_base
[...]	health	location

Offset: 0xC

MEMORY ANALYSIS: ENEMIES



```
f: {...}
  [0]: -118.96875
  [1]: 2299.48975
  [2]: 212.464218
bunnyhop: false
noflash: false
```

```
main()    a.cpp 356:1
```

```cpp
            writeToAddr((void *)force_jump_addr, {0x06});
        }

        if (noflash)
            writeToAddr((void *)player_flash_time_addr, {0, 0, 0, 0, 0, 0, 0, 0});

        //auto loc = getPlayerLocation();
        //std::cout << loc[0] << " " << loc[2] << " " << loc[1] << std::endl;

        auto f = getEntityLocation((void*)0x7f717a529010);

        usleep(200);
    }

    return 0;
}
```

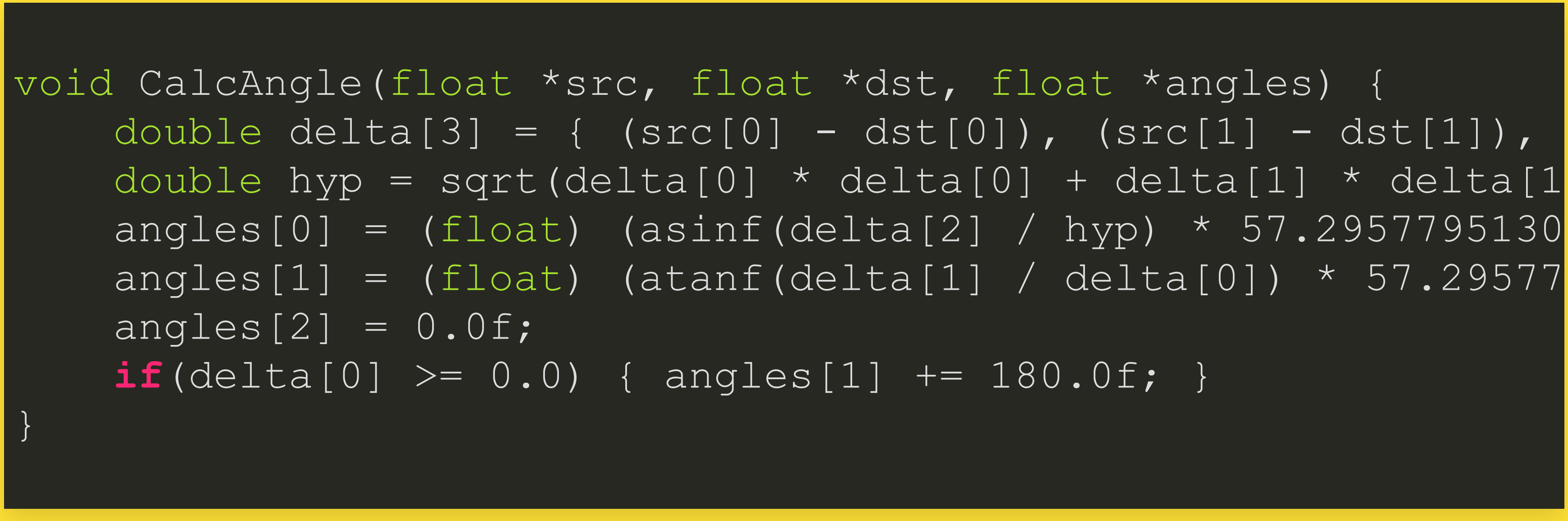
```
force_jump is @ 7f72016bccf0
client_state is @ 9b3a008
view_angle_x is @ 9b42ea0
view_angle_y is @ 9b42ea4
glow_manager_ptr is @ 7f72016b1980
glow_manager is @ 7f7180b1f000
```

NEAREST ENEMY

```
std::sqrt(
    std::pow(entity_x - own_x, 2) +
    std::pow(entity_y - own_y, 2) +
    std::pow(entity_z - own_z, 2)
);
```

CALCULATING THE CAMERA ANGLE

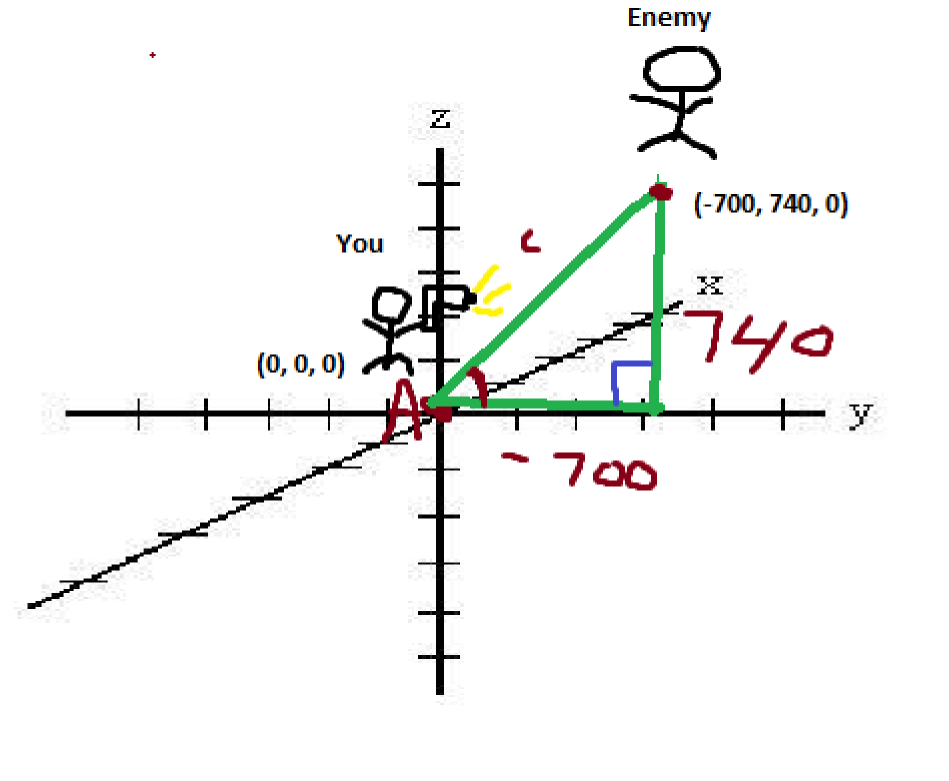
LAST LISTING, I SWEAR



```
void CalcAngle(float *src, float *dst, float *angles) {
    double delta[3] = { (src[0] - dst[0]), (src[1] - dst[1]),
    double hyp = sqrt(delta[0] * delta[0] + delta[1] * delta[1
    angles[0] = (float) (asinf(delta[2] / hyp) * 57.2957795130
    angles[1] = (float) (atanf(delta[1] / delta[0]) * 57.29577
    angles[2] = 0.0f;
    if(delta[0] >= 0.0) { angles[1] += 180.0f; }
}
```

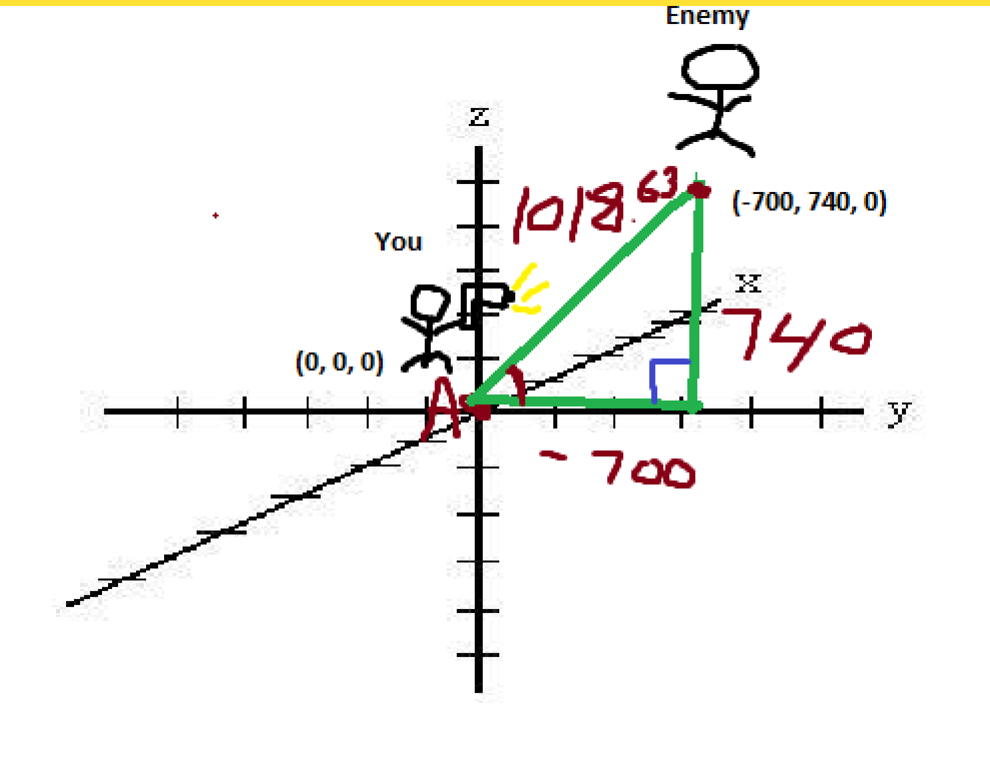
[src]

MAD MATH



[src]

MOAR MATH



[src]

SETTING THE CAMERA ANGLE

- `cl_showpos 1`
- CheatEngine: Freeze value
- Find correct address

DEMO

BOT Bob P T Start (RADIO): Sector clear.

VAC DETECTION

- "VAC is a Joke"
- Uses signatures (among other things)
- Detects specific kinds of hooks
- Solution: Hook mid function
- Don't use public code
- Manual Mapping, Polymorphism and all that fancy malware stuff
- Check out my ROOTCON talk in October for moar on this!

<3

@CaptnBanana

REFERENCES

- My Blog Posts
- Guided Hacking
- UnknownCheats
- idTech3 Engine Analysis
- Random Meme Sites