SIGPwny

SP25 Week 15 • 2025-05-04

Game Hacking 101

Louis Asanaka

Announcements

- End of year Social on Wednesday 5/7!
  - Come to our end of year social to celebrate our graduating members and a year's worth of hard work!

ctf.sigpwny.com

sigpwny{4_r34l_64m3_h4ck3r}

What is Game Hacking?

- Versus cheating
  - Cheating: Gaining unfair advantages in multiplayer games (e.g., aimbots, wallhacks).
  - Modding: Enhancing or changing gameplay for personal enjoyment (e.g., new levels, characters).
  - Research: Analyzing game mechanics, understanding software vulnerabilities.
- Ethical Considerations:
  - Personal Use: Modifying single-player games for fun or accessibility.
  - Online Cheating: Violating terms of service!

Preface

- Please do not cheat in online, multiplayer games!
  - Ruins the fun for others

Target: AssaultCube

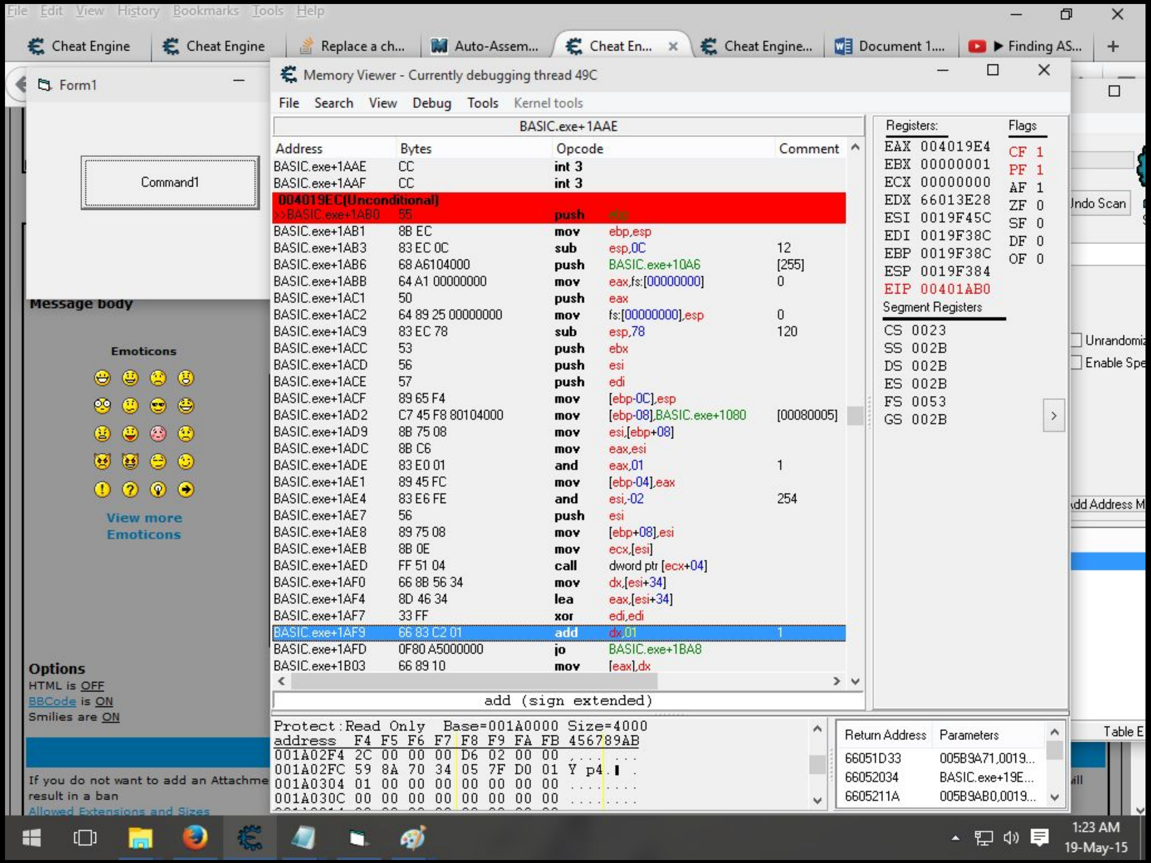
- Cross-platform (although Windows is prevalent)
- Non-Unity game to learn about assembly
  - Unity games are in C#, which are easier to debug & disassemble
- Fun, classic FPS game

External vs Internal

- Refers to memory access (WriteProcessMemory vs pointers)
    - External lives outside of the game in a separate process
    - Internal co-exists with a game
        - Usually requires an injector, a helper to insert your code into the game
- Considerations
    - Efficiency
    - Effectiveness
        - Internals are usually more powerful
    - Anti-cheat detection

Debugging Basics

- Goal is to grasp how the game updates state & networks
  - Inspect variables (e.g. health)
  - Functions that update state
- Works in tandem with reverse engineering!



Memory Viewer - Currently debugging thread 49C

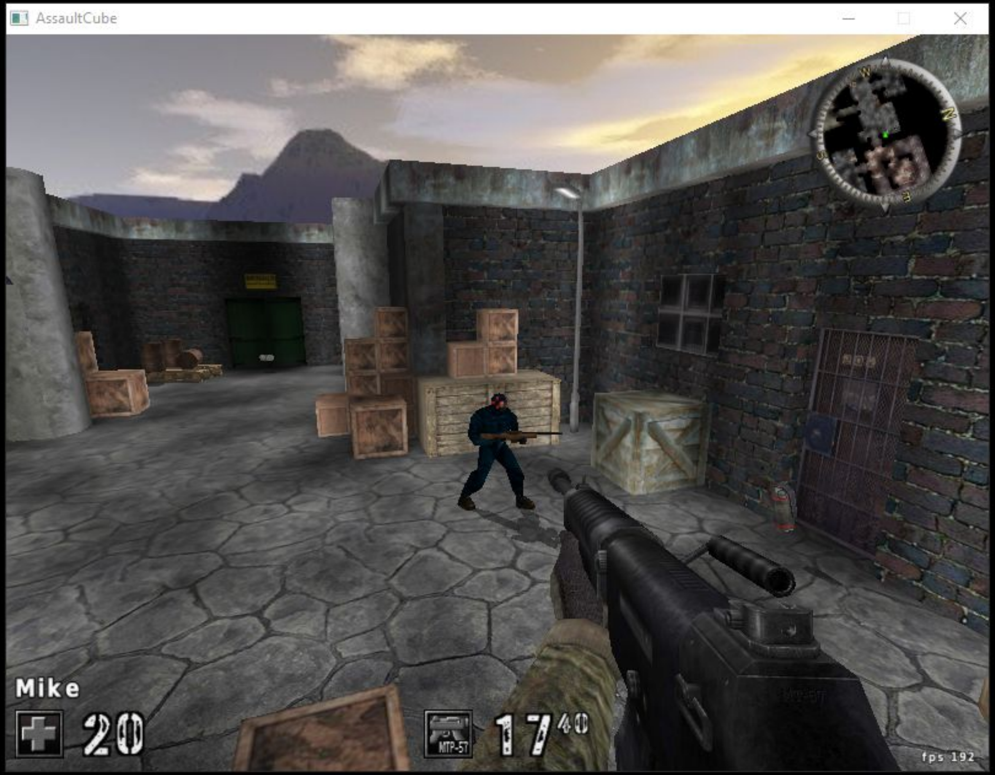
```
BASIC.exe+1AAE
Address            Bytes                 Opcode                              Comment
BASIC.exe+1AAE     CC                    int 3
BASIC.exe+1AAF     CC                    int 3
 004019EC(Unconditional)
>>BASIC.exe+1AB0   55                    push   ebp
BASIC.exe+1AB1     8B EC                 mov    ebp,esp
BASIC.exe+1AB3     83 EC 0C              sub    esp,0C                       12
BASIC.exe+1AB6     68 A6104000           push   BASIC.exe+10A6               [255]
BASIC.exe+1ABB     64 A1 00000000        mov    eax,fs:[00000000]            0
BASIC.exe+1AC1     50                    push   eax
BASIC.exe+1AC2     64 89 25 00000000     mov    fs:[00000000],esp            0
BASIC.exe+1AC9     83 EC 78              sub    esp,78                       120
BASIC.exe+1ACC     53                    push   ebx
BASIC.exe+1ACD     56                    push   esi
BASIC.exe+1ACE     57                    push   edi
BASIC.exe+1ACF     89 65 F4              mov    [ebp-0C],esp
BASIC.exe+1AD2     C7 45 F8 80104000     mov    [ebp-08],BASIC.exe+1080      [00080005]
BASIC.exe+1AD9     8B 75 08              mov    esi,[ebp+08]
BASIC.exe+1ADC     8B C6                 mov    eax,esi
BASIC.exe+1ADE     83 E0 01              and    eax,01                       1
BASIC.exe+1AE1     89 45 FC              mov    [ebp-04],eax
BASIC.exe+1AE4     83 E6 FE              and    esi,-02                      254
BASIC.exe+1AE7     56                    push   esi
BASIC.exe+1AE8     89 75 08              mov    [ebp+08],esi
BASIC.exe+1AEB     8B 0E                 mov    ecx,[esi]
BASIC.exe+1AED     FF 51 04              call   dword ptr [ecx+04]
BASIC.exe+1AF0     66 8B 56 34           mov    dx,[esi+34]
BASIC.exe+1AF4     8D 46 34              lea    eax,[esi+34]
BASIC.exe+1AF7     33 FF                 xor    edi,edi
BASIC.exe+1AF9     66 83 C2 01           add    dx,01                        1
BASIC.exe+1AFD     0F80 A5000000         jo     BASIC.exe+1BA8
BASIC.exe+1B03     66 89 10              mov    [eax],dx

                    add (sign extended)
```

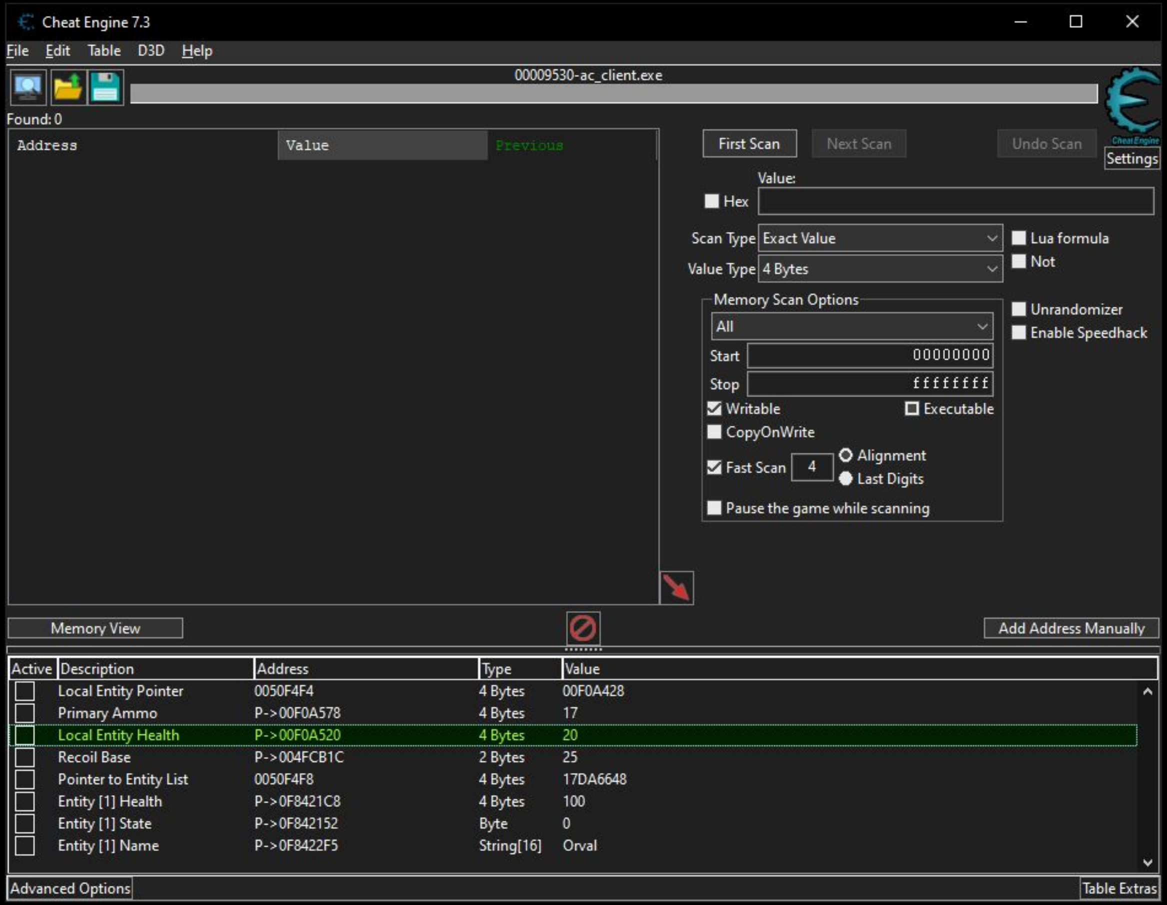
```
Registers:            Flags
EAX 004019E4          CF 1
EBX 00000001          PF 1
ECX 00000000          AF 1
EDX 66013E28          ZF 0
ESI 0019F45C          SF 0
EDI 0019F38C          DF 0
EBP 0019F38C          OF 0
ESP 0019F384
EIP 00401AB0
Segment Registers
CS 0023
SS 002B
DS 002B
ES 002B
FS 0053
GS 002B
```

```
Protect:Read Only  Base=001A0000 Size=4000
address  F4 F5 F6 F7 F8 F9 FA FB 456789AB
001A02F4 2C 00 00 00 D6 02 00 00 ,.......
001A02FC 59 8A 70 34 05 7F D0 01 Y p4.▌ .
001A0304 01 00 00 00 00 00 00 00 ........
001A030C 00 00 00 00 00 00 00 00 ........
```

Return Address	Parameters
66051D33	005B9A71,0019...
66052034	BASIC.exe+19E...
6605211A	005B9AB0,0019...

AssaultCube & Cheat Engine





00009530-ac_client.exe

Found: 0

Address	Value	Previous

Value:

☐ Hex

Scan Type: Exact Value

Value Type: 4 Bytes

☐ Lua formula

☐ Not

Memory Scan Options: All

Start: 00000000

Stop: ffffffff

☑ Writable ☐ Executable

☐ CopyOnWrite

☑ Fast Scan 4 ○ Alignment ● Last Digits

☐ Pause the game while scanning

☐ Unrandomizer

☐ Enable Speedhack

Active	Description	Address	Type	Value
☐	Local Entity Pointer	0050F4F4	4 Bytes	00F0A428
☐	Primary Ammo	P->00F0A578	4 Bytes	17
☐	Local Entity Health	P->00F0A520	4 Bytes	20
☐	Recoil Base	P->004FCB1C	2 Bytes	25
☐	Pointer to Entity List	0050F4F8	4 Bytes	17DA6648
☐	Entity [1] Health	P->0F8421C8	4 Bytes	100
☐	Entity [1] State	P->0F842152	Byte	0
☐	Entity [1] Name	P->0F8422F5	String[16]	Orval

Modifying Code

- Basic Workflow
  - Directly altering game binaries
    - Checksums in the game can detect this
  - Dynamically altering instructions in a debugger
- Example
  - Bypassing checks: Changing conditionals to skip over health checks
  - Changing parameters: Damage * 100 instead of 2
  - Generic code injection: Anything you want!

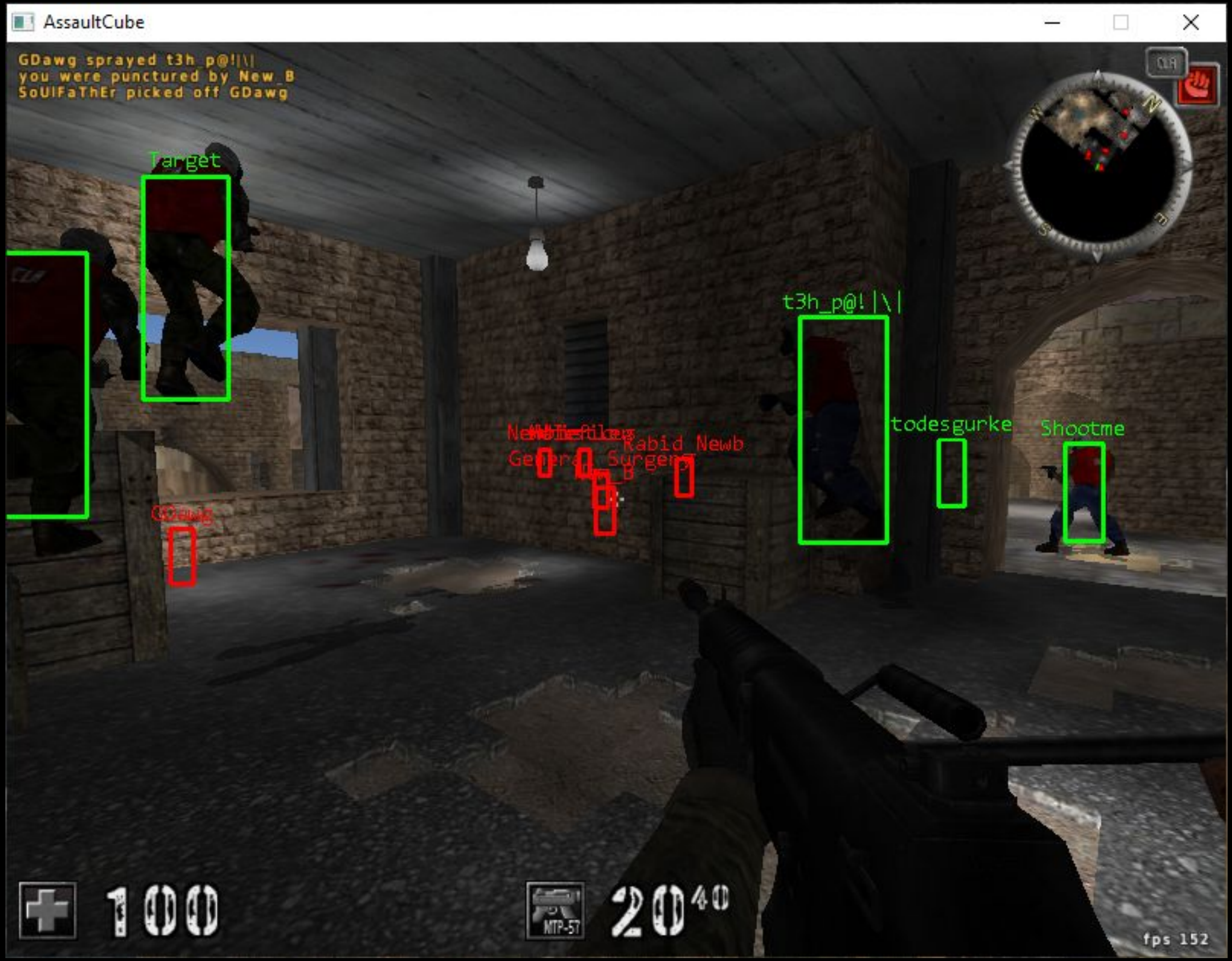
Persisting Manual Changes

- Time to write code to automatically do things!
- Languages of choice: C / C++
  - Ease of interacting with raw memory
  - Python/C# is also common for external cheats
- Internal cheats need to be loaded – how?

DLL / Library Injection

- TL;DR: Getting your code running within the game
- DLL = Windows library format
  - .so on Linux, .dylib in MacOS
  - But no one games on those platforms :)
- Example Methods
  - LoadLibrary: A Windows API function used to load DLLs.
  - CreateRemoteThread: A Windows API function used to execute code in another process

Internal Memory Read + Graphics



AssaultCube

GDawg sprayed t3h_p@!|\|
you were punctured by New_B
SoUlFaThEr picked off GDawg

Obstacles: Basic Anti-cheat

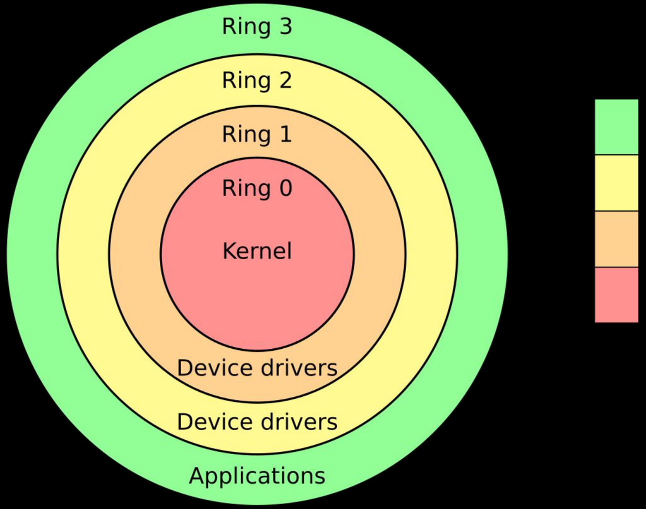
- Checksums
  - Hash for game files to ensure integrity
- Value checks
  - Server checks for money, health, etc.
- Pattern scanning
  - Game scans its own memory for suspicious code / known cheats
- Behavior monitoring
  - E.g. Minecraft limits player reach to reasonable values

Obstacles: Basic Anti-cheat

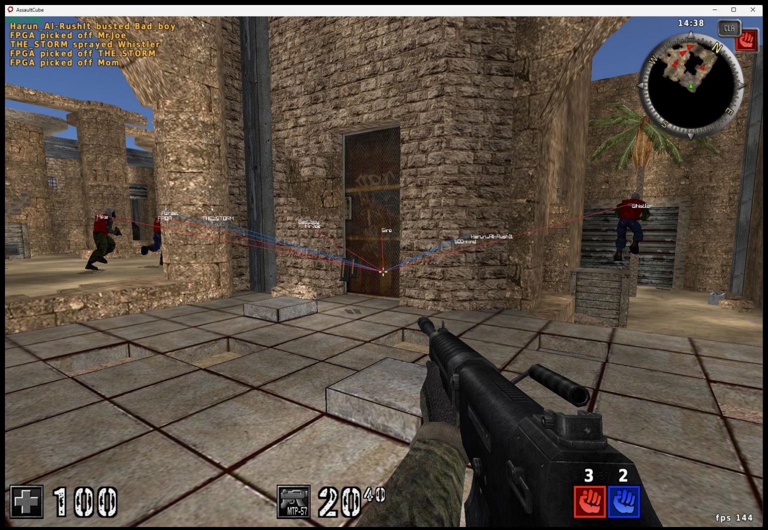
- Code obfuscation
  - Things like VMProtect to make control flow impossible to read
  - Trade-off with performance for the game developer!
- Debugger traps
  - Try to stop you from attaching a debugger

Obstacles: Modern Anti-cheat

- Kernel anti-cheat
  - Regular applications can’t see the anti-cheat!
  - E.g. EasyAntiCheat, Vanguard
- ML-based behavior analysis
  - Cheating behaviors are detected over-time
  - E.g. VACNet

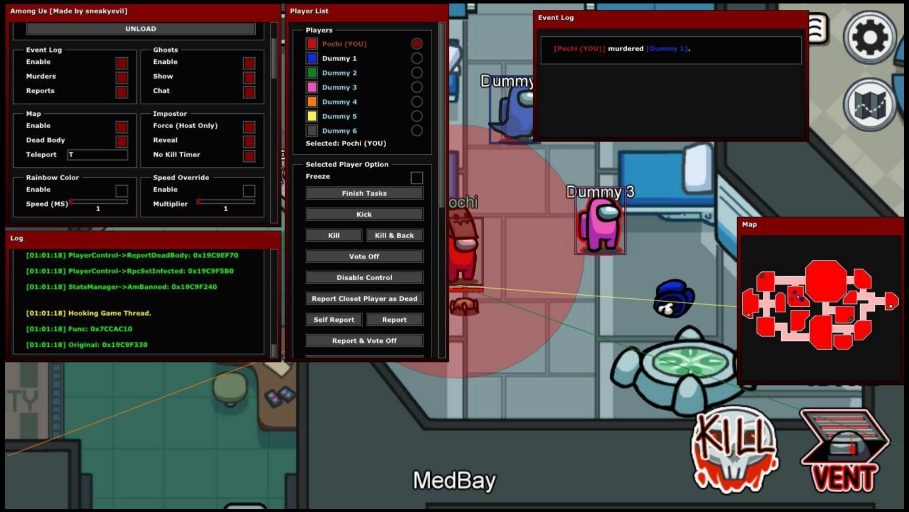


Challenges



```
Harun_Al-RushIt busted Bad_boy
FPGA picked off MrJoe
THE_STORM sprayed Whistler
FPGA picked off THE_STORM
FPGA picked off Mom
```

- AssaultCube has no anti-cheat
  - Anything is possible!
  - E.g. Global god mode, perfect aim, infinite bullets etc.
- Unity Games
  - Typically are indie and have little anti-cheat if any
  - E.g. Among Us



```
[01:01:18] PlayerControl->ReportDeadBody: 0x19C9EF70
[01:01:18] PlayerControl->RpcSetInfected: 0x19C9F5B0
[01:01:18] StatsManager->AmBanned: 0x19C9F240

[01:01:18] Hooking Game Thread.
[01:01:18] Func: 0x7CCAC10
[01:01:18] Original: 0x19C9F330
```

[Pochi (YOU)] murdered [Dummy 1].

Resources

- Cheat Engine (Website)
  - General purpose debugger with convenient memory tools
- dotPeek (Website)
  - Free .NET (C#) Decompiler and Assembly Browser
- AssaultCube (Website)
  - Super simple cross-platform FPS
- GuidedHacking (Website)
  - Guides ranging from Cheat Engine to binary exploitation
- GameHacking Academy (Website)
  - Comprehensive course

Next Meetings

2025-05-07 • **This Wednesday**

- End of year Social
  - Come to our end of year social to celebrate our graduating members and a year's worth of hard work!

ctf.sigpwny.com

sigpwny{4_r34l_64m3_h4ck3r}

Meeting content can be found at sigpwny.com/meetings.

SIGPwny