

RSA® Conference 2018

San Francisco | April 16–20 | Moscone Center

SESSION ID: HTA-R04

ANATOMY OF EXPLOITING MMORPG'S

Adrian Bednarek

Security Analyst/Researcher
Independent Security Evaluators
@ISEsecurity





Obligatory who is this guy?

- Adrian Bednarek
- Security Analyst/Researcher at ISE (Independent Security Evaluators)
- Started in the security field as a mostly ethical blackhat*
 - Creating side channel ‘in app purchase’ functionality
 - E.g. Selling gold and virtual goods on ebay, then playerauctions/wholesale
 - First sale – castle in Ultima Online
 - Over 100+ 0day virtual economy exploits in 24+ online games
- Here to talk about common exploits in MMORPG’s
 - Tools
 - Methods
 - Challenges!

What is an MMORPG?



- Massively Multiplayer Online Role Playing Game
 - Single virtual world with multiple thousands of players interacting
- Typically used to be windows based
 - MMORPG's are highly popular on iOS/Android

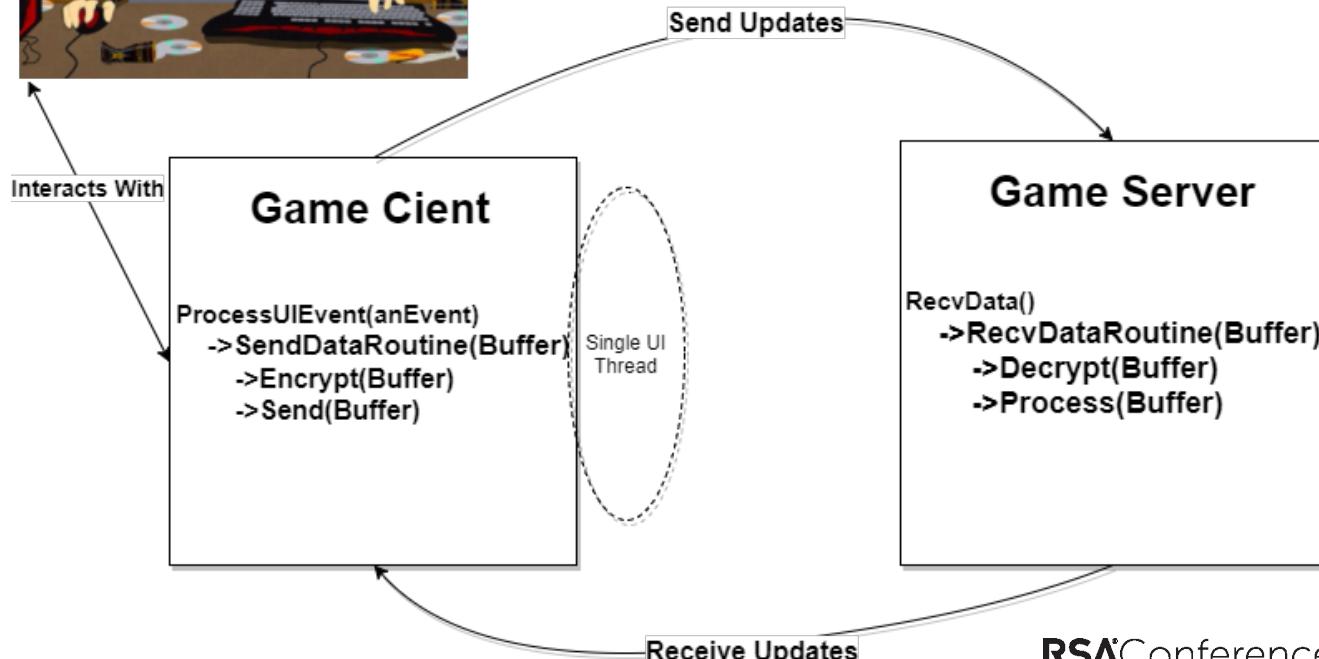
ESO



<http://www.justpushstart.com/2014/04/elder-scrolls-online-guide-banished-cells-dungeon-overview/>

RSA Conference 2018

Overly simplified game architecture

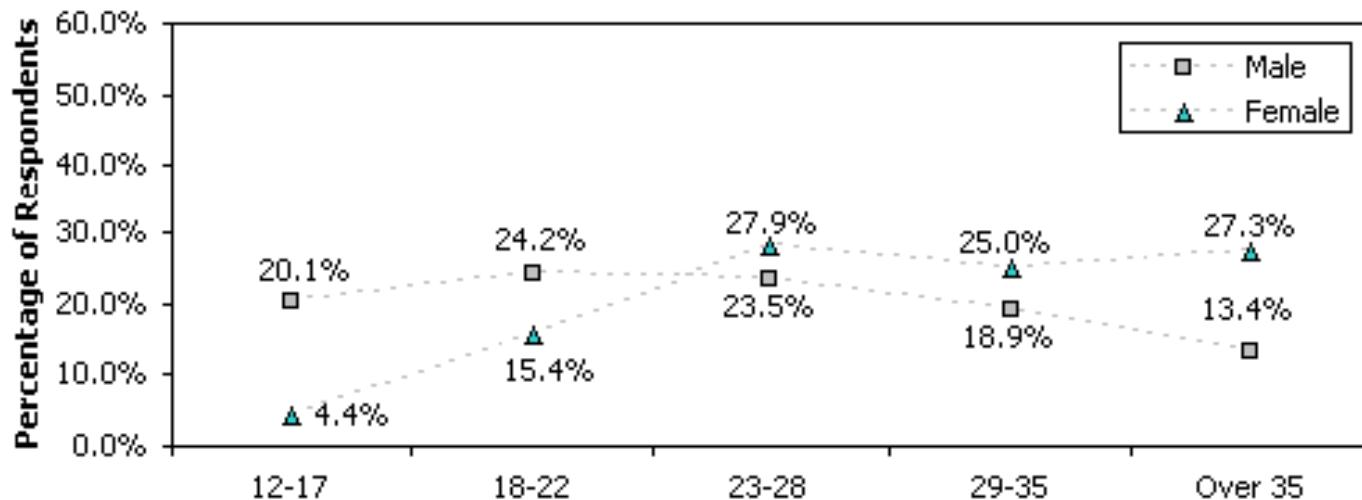


Player Demographics



Gender and Age Distribution

N male = 2439, N female = 404

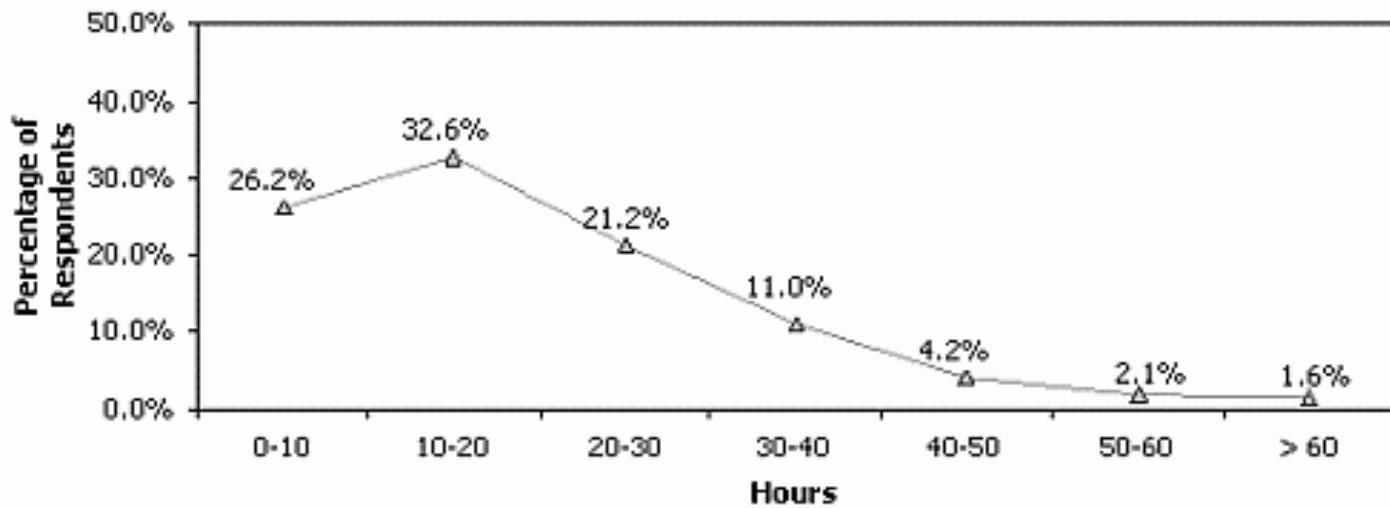


https://www.gamasutra.com/view/feature/130552/unmasking_the_avatar_the_.php

Player Demographics



Hours Per Week Distribution
N = 2982



https://www.gamasutra.com/view/feature/130552/unmasking_the_avatar_the_.php

MMORPG Popularity



- Lots of players across many demographics!

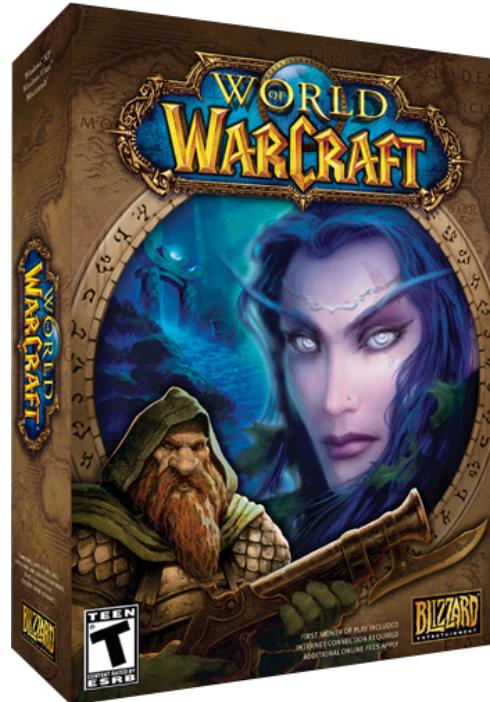


<https://inage.com/2015/11/04/blizzards-q3-2015-report/>

MMORPG Business Model



- Early Days (1990s to early 2010 years)
 - Subscription model
 - Buy a box
 - Includes 30 days of playtime
 - To continue playing you typically pay 15\$ per month
 - Unlimited unrestricted play (all you can eat!)
 - Seldomly used these days



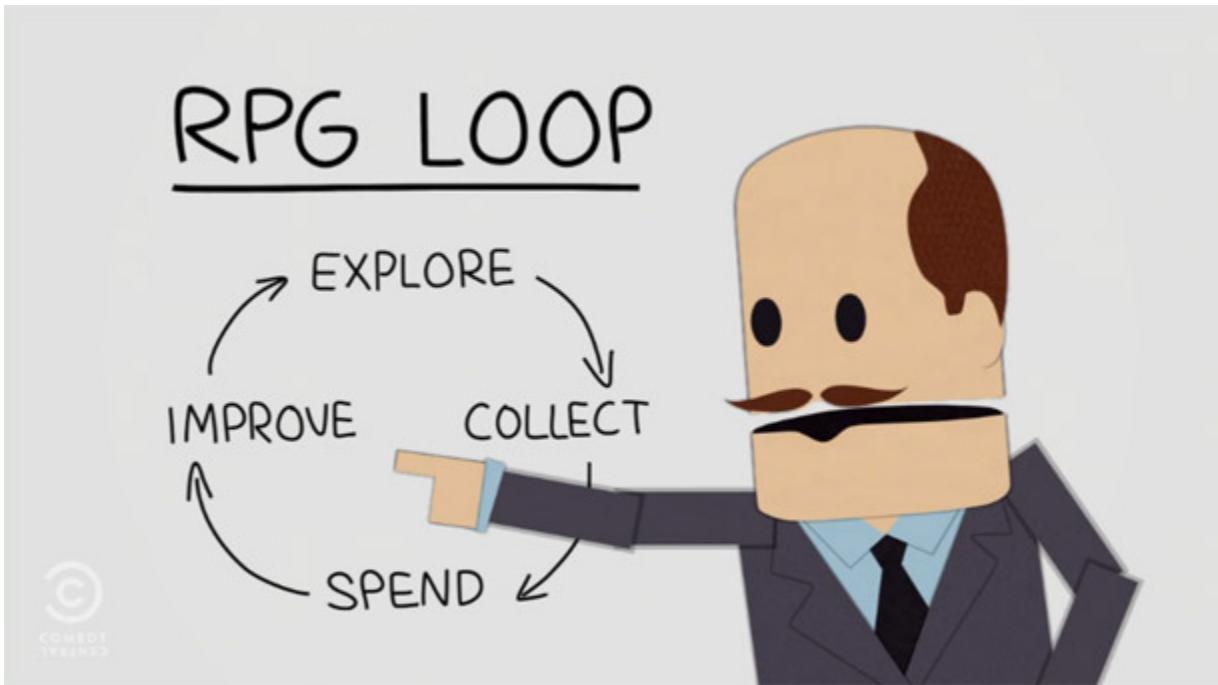
MMORPG Business Model



- Current
 - Free/One time fee for game
 - In app purchase model
 - Buy items that enhance gameplay
 - 'Pay 2 Win' complaints
 - Cosmetic items
 - Play time limit
 - To continue playing pay more
 - Or wait! (maybe*)

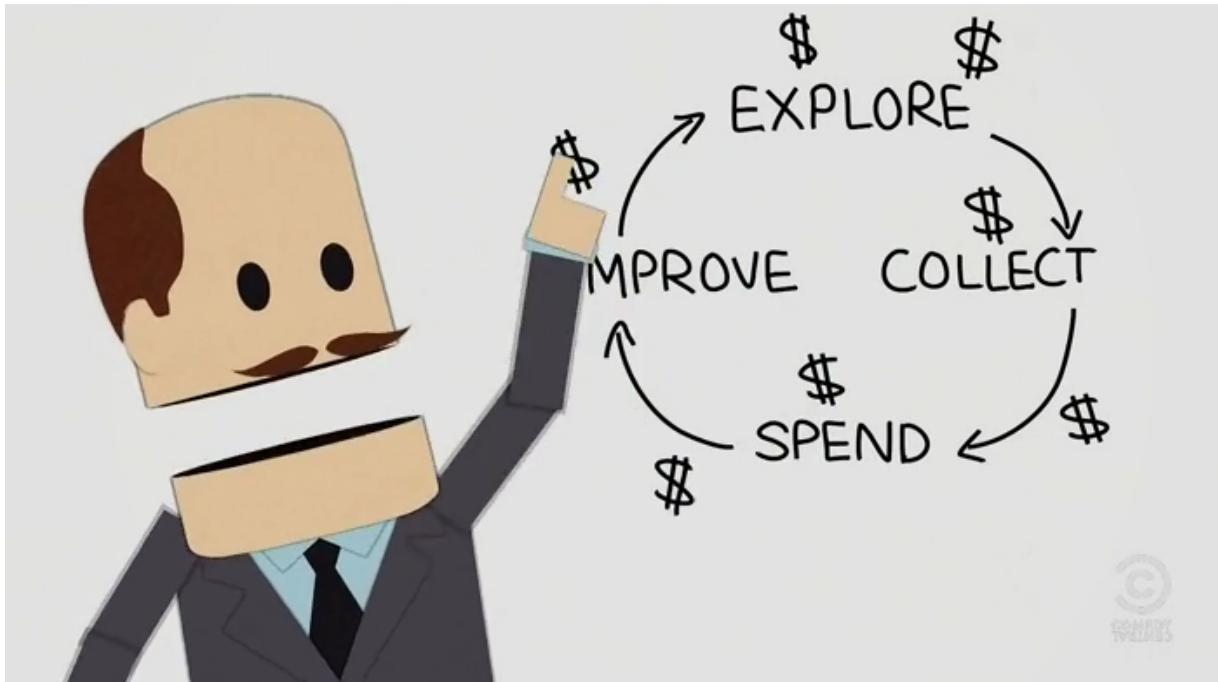


MMORPG Business Model



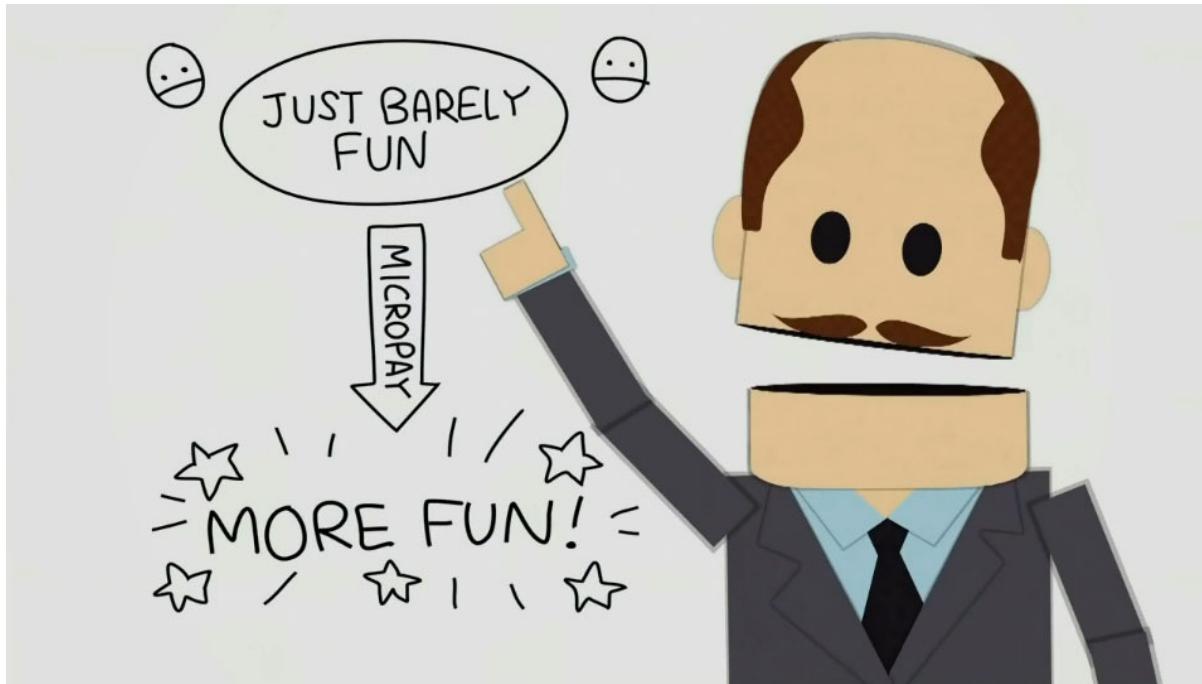
*South Park Studios – Freemium Isn't Free

MMORPG Business Model



*South Park Studios – Freemium Isn't Free

MMORPG Business Model



*South Park Studios – Freemium Isn't Free

Pay for virtual game goods?



- Paying for virtual game goods
 - That's crazy talk!
 - Why not just hack it?
- But first... Integer overflows!





Integer Overflows!

The screenshot shows the Immunity Debugger interface with the CPU tab selected. The assembly window displays the following code:

```
CC int3
90 nop
C3 ret
8D A4 24 00 00 00 00 lea esp,dword ptr ss:[esp]
8D 9B 00 00 00 00 lea ebx,dword ptr ds:[ebx]
90 nop
B8 FF FF FF FF mov eax,FFFFFF
83 C0 01 add eax,1
90 nop
90 nop
2B D8 sub ebx,eax
2B D8 sub ebx,eax
2B D8 sub ebx,eax
F6 41 04 06 test byte ptr ds:[ecx+4],6
74 05 je ntdll.772707AF
E8 E1 F8 FF FF call <ntdll.NtTestAlert>
```

The EIP register is highlighted with a blue arrow pointing to the instruction at address 77270791.

The Registers window shows the following values:

| | Hide FPU | |
|--------|-------------------------|----------------|
| EAX | 00000000 | |
| EBX | FFFFFFFF | |
| ECX | 772A40C0 <ntdll.dbgUIRe | |
| EDX | 772A40C0 <ntdll.dbgUIRe | |
| EBP | 007FFF80 | |
| ESP | 007FFF54 | |
| ESI | 772A40C0 <ntdll.dbgUIRe | |
| EDI | 772A40C0 <ntdll.dbgUIRe | |
| EIP | 77270791 | ntdll.77270791 |
| EFLAGS | 00000295 | |
| ZF | 0 | PF 1 AF 1 |
| OF | 0 | SE 1 DF 0 |
| CF | 1 | TF 0 IF 1 |



Integer Overflows!

The screenshot shows the Immunity Debugger interface with the CPU tab selected. The assembly window displays the following code:

```
CC int3
90 nop
C3 ret
8D A4 24 00 00 00 00 lea esp,dword ptr ss:[esp]
8D 9B 00 00 00 00 lea ebx,dword ptr ds:[ebx]
90 nop
B8 FF FF FF FF mov eax,FFFFFF
83 C0 01 add eax,1
90 nop
90 nop
2B D8 sub ebx,eax
2B D8 sub ebx,eax
2B D8 sub ebx,eax
90 nop
90 nop
90 nop
F6 41 04 06 test byte ptr ds:[ecx+4],6
74 05 je ntdll.772707AF
E8 E1 F8 FF FF call <ntdll.NtTestAlert>
```

The instruction at address 77270796 (opcode 83 C0 01) is highlighted. The Registers window on the right shows the following register values:

| Hide FPU | | | |
|----------|----------|----------------|------|
| EAX | FFFFFFFF | | |
| EBX | FFFFFFF | | |
| ECX | 772A40C0 | <ntdll.dbgUiRe | |
| EDX | 772A40C0 | <ntdll.dbgUiRe | |
| EBP | 007FFF80 | | |
| ESP | 007FFF54 | | |
| ESI | 772A40C0 | <ntdll.dbgUiRe | |
| EDI | 772A40C0 | <ntdll.dbgUiRe | |
| EIP | 77270796 | ntdll.77270796 | |
| EFLAGS | 00000297 | | |
| ZF | 0 | PF 1 | AF 1 |
| OF | 0 | SE 1 | DF 0 |
| CF | 1 | TF 0 | IF 1 |



Integer Overflows!

32 CPU Graph Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols

| | 77270780 | CC |
|-----|----------|----------------------|
| | 77270781 | 90 |
| | 77270782 | C3 |
| | 77270783 | 8D A4 24 00 00 00 00 |
| | 7727078A | 8D 9B 00 00 00 00 |
| | 77270790 | 90 |
| | 77270791 | B8 FF FF FF FF |
| | 77270796 | 83 C0 01 |
| EIP | 77270799 | 90 |
| | 7727079A | 90 |
| | 7727079B | 2B D8 |
| | 7727079D | 2B D8 |
| | 7727079F | 2B D8 |
| | 772707A1 | 90 |
| | 772707A2 | 90 |
| | 772707A3 | 90 |
| | 772707A4 | F6 41 04 06 |
| | 772707A8 | 74 05 |
| | 772707AA | E8 E1 F8 FF FF |

int3
nop
ret
lea esp,dword ptr ss:[esp]
lea ebx,dword ptr ds:[ebx]
nop
mov eax,FFFFFFFF
add eax,1
nop
nop
sub ebx,eax
sub ebx,eax
sub ebx,eax
nop
nop
nop
test byte ptr ds:[ecx+4],6
je ntdll.772707AF
call <ntdll.NtTestAlert>

Hide FPU

| EAX | 00000000 | |
|--------|----------|----------------|
| EBX | FFFFFFFF | <ntdll.dbgUiRe |
| ECX | 772A40C0 | <ntdll.dbgUiRe |
| EDX | 772A40C0 | <ntdll.dbgUiRe |
| EBP | 007FFF80 | |
| ESP | 007FFF54 | |
| ESI | 772A40C0 | <ntdll.dbgUiRe |
| EDI | 772A40C0 | <ntdll.dbgUiRe |
| EIP | 77270799 | ntdll.77270799 |
| EFLAGS | 00000257 | |
| ZF | 1 | PF 1 AF 1 |
| OF | 0 | SF 0 DF 0 |
| CF | 1 | TF 0 IF 1 |



Pay for virtual game goods?

- An integer overflow in the wild:
 - [Demo: WSO.flv](#)



Pay for virtual game goods?

- Submit a buy order for max signed int64
 - 7FFFFFFF FFFFFFFF
 - Server adds a small fee
 - 7FFFFFFF FFFFFFFF + fee = integer roll into negative
 - >8000000 00000000
 - This then gets subtracted from the player account balance

| | |
|-----------------|-------------------|
| 006FF6D8 | 000006CA |
| 006FF6DC | 006FF6F0 |
| 006FF6E0 | 006FFBCB |
| 006FF6E4 | OE445 810 |
| 006FF6E8 | 006FFBCB |
| 006FF6EC | CD5B2523 |
| 006FF6F0 | 00000000 |
| 006FF6F4 | 00000000 |
| 006FF6F8 | 0000490A |
| 006FF6FC | 00000001 |
| 006FF700 | FFFFFFFFFF |
| 006FF704 | 7FFFFFFF |
| 006FF708 | 00000006 |
| 006FF70C | 00000000 |
| 006FF710 | 00000001 |
| 006FF714 | 00000000 |
| 006FF718 | 00000000 |
| 006FF71C | 00000000 |
| 006FF720 | 00000000 |

Pay for virtual game goods?

- Subtracting a negative number puts you in the positive
 - e.g. $5 - -8 = 13$
- $9.038904600371e+018$
- 9 Quintillion gold! (credits)
- (Implications later)

Player money Window:



21:29 [Loot] : You receive -2001557840 Platinum, 84 Gold, 88 Silver, 97 Copper
21:30 [Debug] : 9.038904600371e+018

Log + Debug command to show player money

RSA® Conference 2018



#RSAC

MALICIOUS ACTORS

Hacking MMORPGs





But first...Motivation?

- 1) For the lulz:

lulz

/ləlz/

noun informal

fun, laughter, or amusement, especially that derived at another's expense.

"the splinter group embarked on a spree of daring cyberattacks **for the lulz**"

But first...Motivation?



- 2) For profit:



Type 1) Adversaries/Researchers

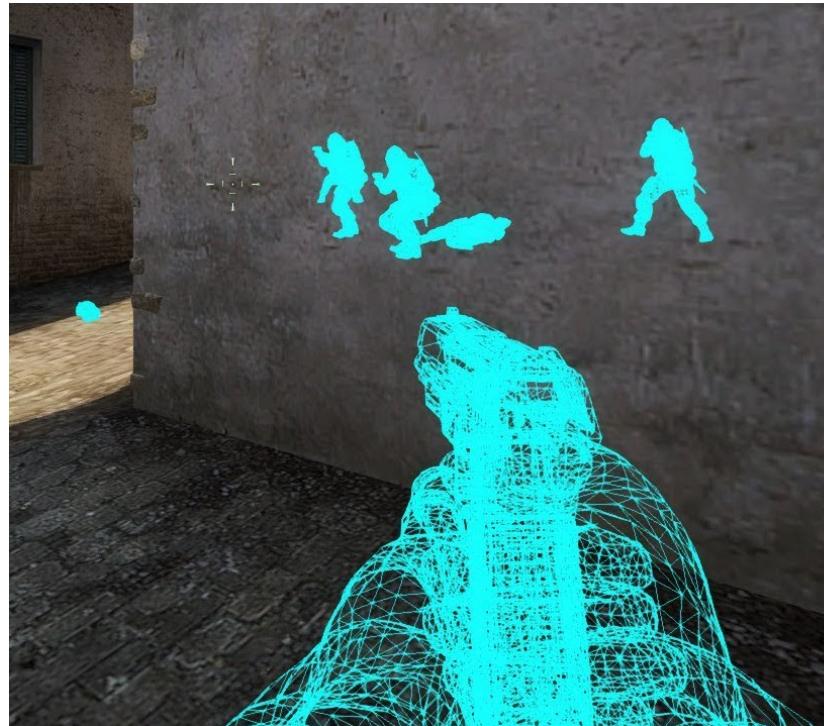
- Researchers
 - Can I bypass client side checks
 - What kind of encryption is being used
 - What libraries are being used?
 - Can I view art/sound files?
 - How does the protocol work
 - How do they validate inputs



Type 2) Adversaries/Hackers



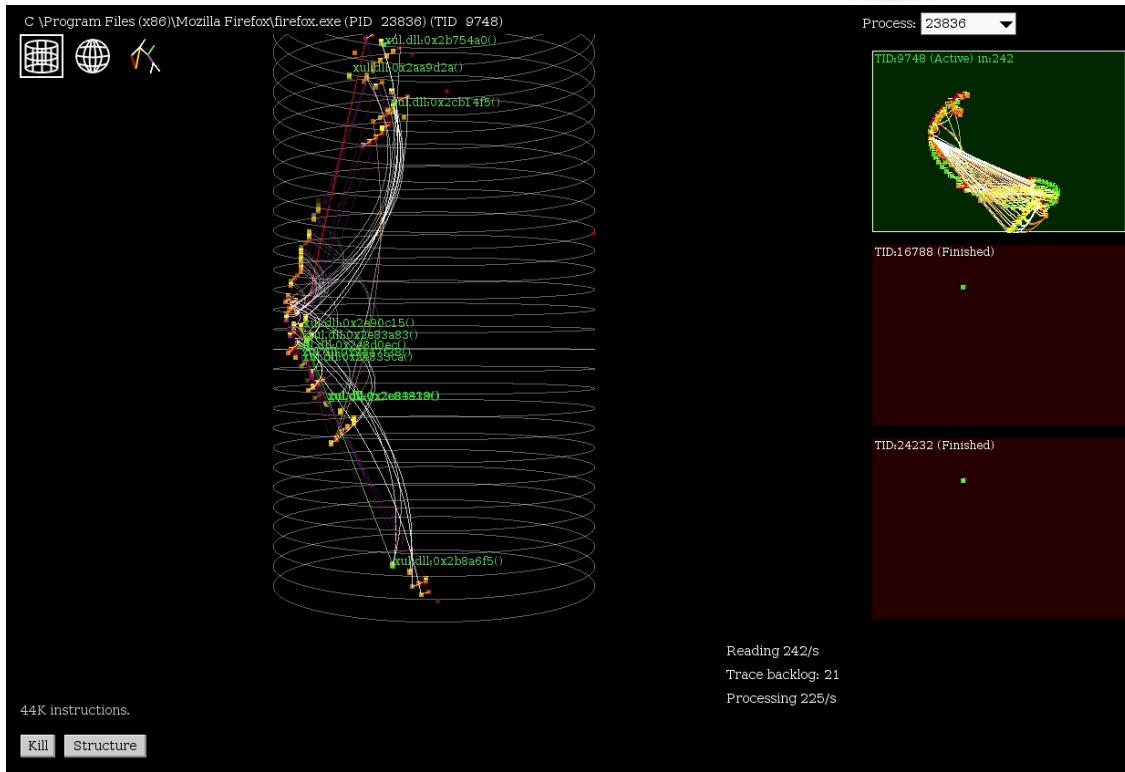
- Cheaters/Hackers (for the advantage)
 - Client RE
 - Modify game client
 - Runtime
 - Static
 - Typically to gain advantage over players
 - Disruptive
 - Upset players (ever been 360 noscoped?)



Type 3) Adversaries/Professional Hackers



- For profit
 - In game commodities
 - Tools/Bots
 - Custom
 - Off shelf
 - 0-Days
 - Server compromise
 - Client compromise
 - Countermeasure evasion
 - High level statistics
 - Client modification



44K instructions.

Kill Structure

*<https://github.com/ncatlin/rgat>

Type 4) Adversaries/Nation states



- Nation states? (*In my game?... its more likely than you think.*)
- North Korea
 - employing MMO hackers to fund government
- South Korea
 - 30 North Korean operatives
 - Bot and farm 6 million USD\$ worth of digital goods (L2)



Adversaries/Nation states

- Chinese prisoners
 - Forced to farm World of Warcraft gold
 - 300 prisoners play WoW 12 hours a day
 - Generate ~800 USD\$ per day



Type 5) Adversaries/Chinese Farmers



- Over 100k active farmers
- Typically
 - Bot assisted
 - Manual
- Sweatshop like conditions
- Detrimental to gameplay
 - Spawn camping
 - Resource sniping



Hacking for profit?



- What a million looks like in USD\$
- Weighs about 22 lbs.



Hacking for profit



- What a million USD\$ looks like in Elder Scrolls Online Currency
 - < billionth of a billionth of a gram, or 0.0000000000000001g*

Hacking for profit



- 1 million ESO gold sells for about 77\$

< Home > Shopping Cart

Shopping Cart

Choose Your Currency **USD** ▾

| Server and Product(s) | Price | Qty | Amount | Remove |
|----------------------------------|-------------|--------------------------------|-------------|--------|
| PC-North America 1000 K ESO Gold | USD\$ 76.99 | <input type="text" value="1"/> | USD\$ 76.99 | |

Total Amount: **USD\$ 76.99**

[<< Continue Add Items](#) [Clean Cart](#) [Proceed to Check Out ➔](#)



Hacking for profit?

- 77\$ per million ESO gold
- $1,000,000/77 = 12,987$ million units of ESO gold
- 12,987,000,000 gold
- That seems like a lot of gold!
- How easy would it be to get that much?
 - Manually – forever
 - Hacking...

5 LEVEL



Remember WildStar?



- 5\$ per million units of WildStar gold

< Home > Shopping Cart

Choose Your Currency **USD** ▾

| Server and Product(s) | Price | Discount | Qty | Amount | Remove |
|--|-------------------------------------|-----------|--------------------------------|------------|--------|
| NA-Entity-Dominion(PVE) 1000 WildStar Gold | USD\$ 5.75 USD\$ 5.18 | (10% OFF) | <input type="text" value="1"/> | USD\$ 4.07 | X |

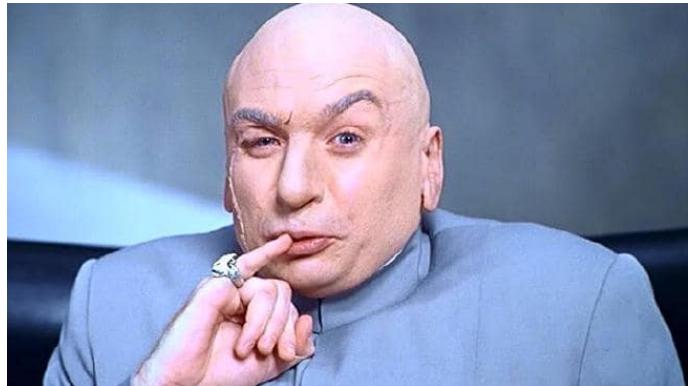
Total Amount: **USD\$ 4.07**

[<< Continue Add Items](#) [Clean Cart](#) [Proceed to Check Out ➔](#)

Hacking for profit?



- 5\$ per million WSO gold
- ~ 9 Quintillion gold
- $(9038904600371000000/1,000,000)*5 = 45,194,523,001,855$ USD\$
- 45 Trillion Dollars

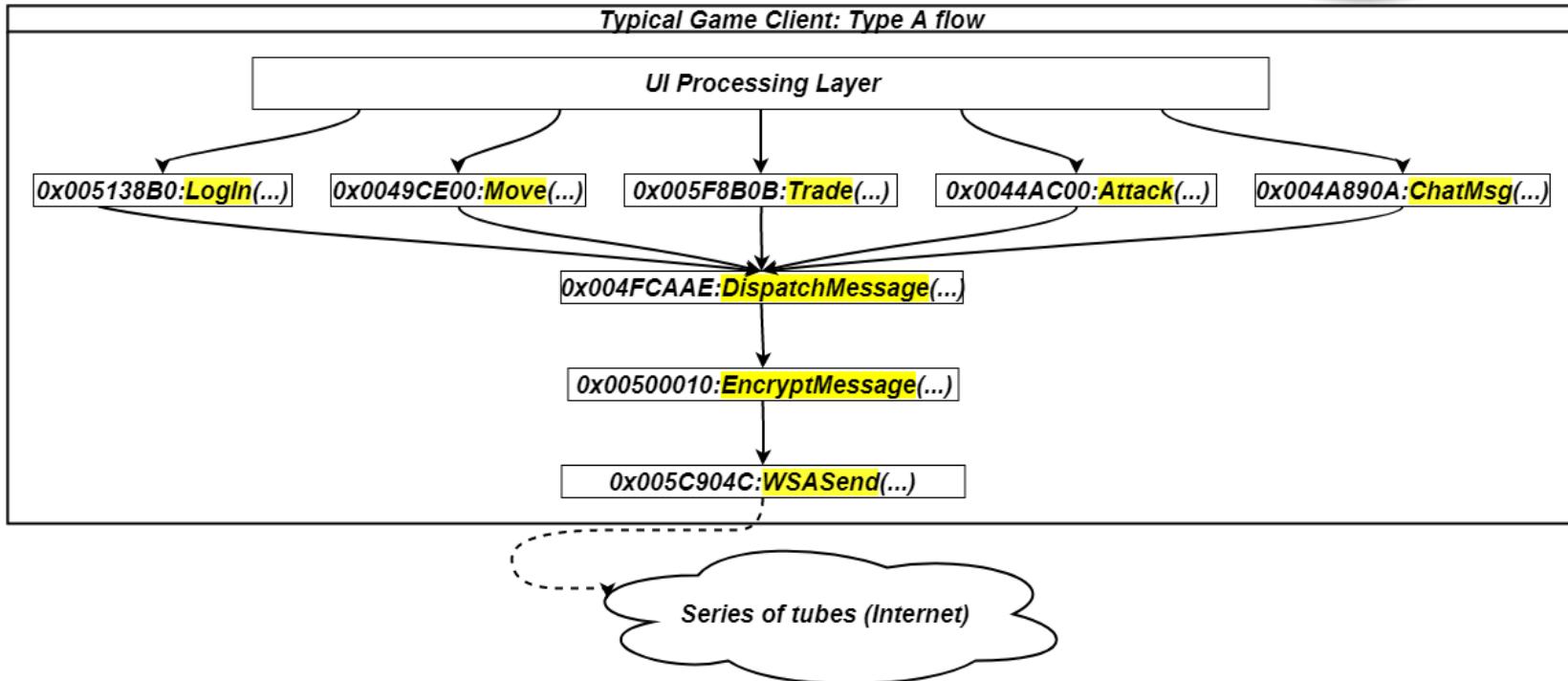




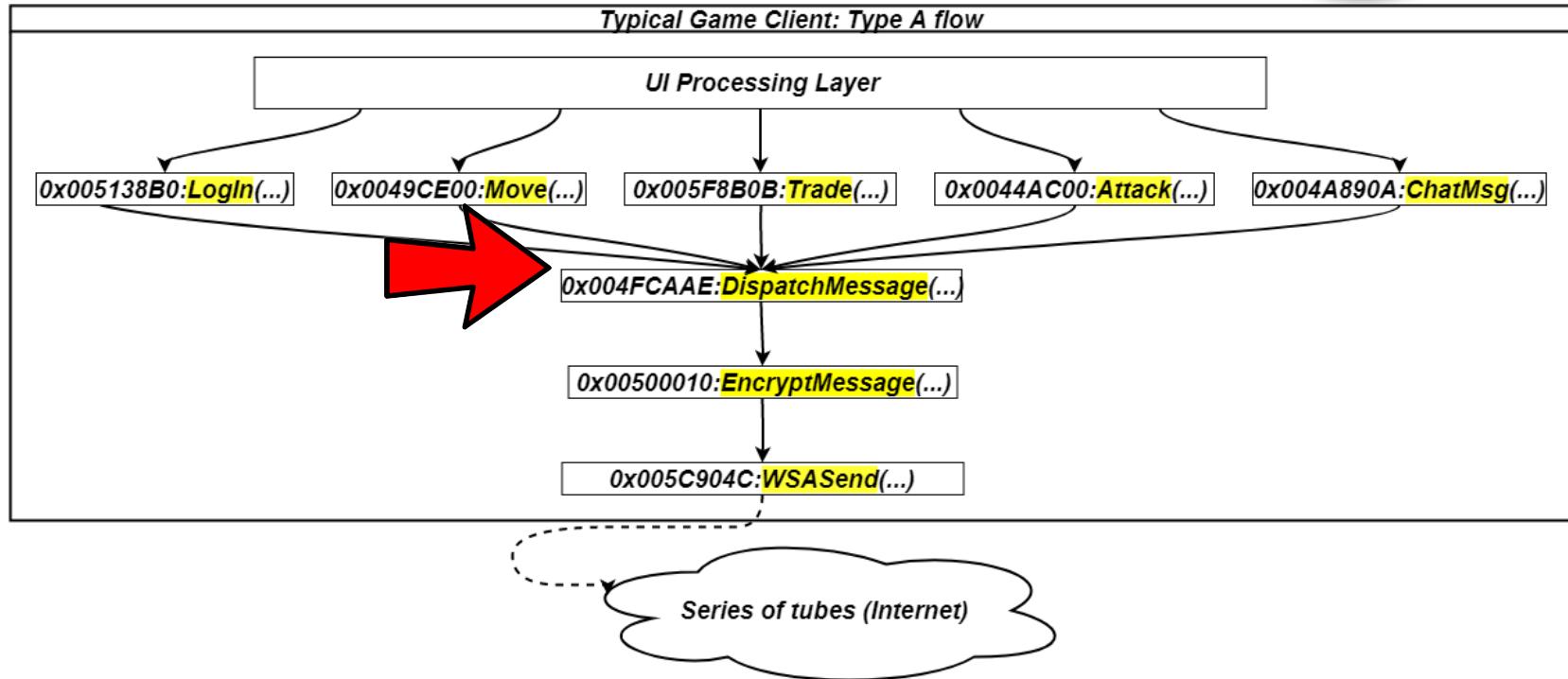
Challenges

- Countermeasures
 - Then – Nearly none
 - Now – Many! (Advanced!)
 - Proprietary encryption
 - Anti Debugging
 - Packet integrity checks
 - Packet sequencing
 - Server side statistics (e.g. Sudden jumps in player wealth)
 - Client/Host machine fingerprinting
 - **Code flow obfuscation!**
 - Advanced countermeasures help
 - adversaries

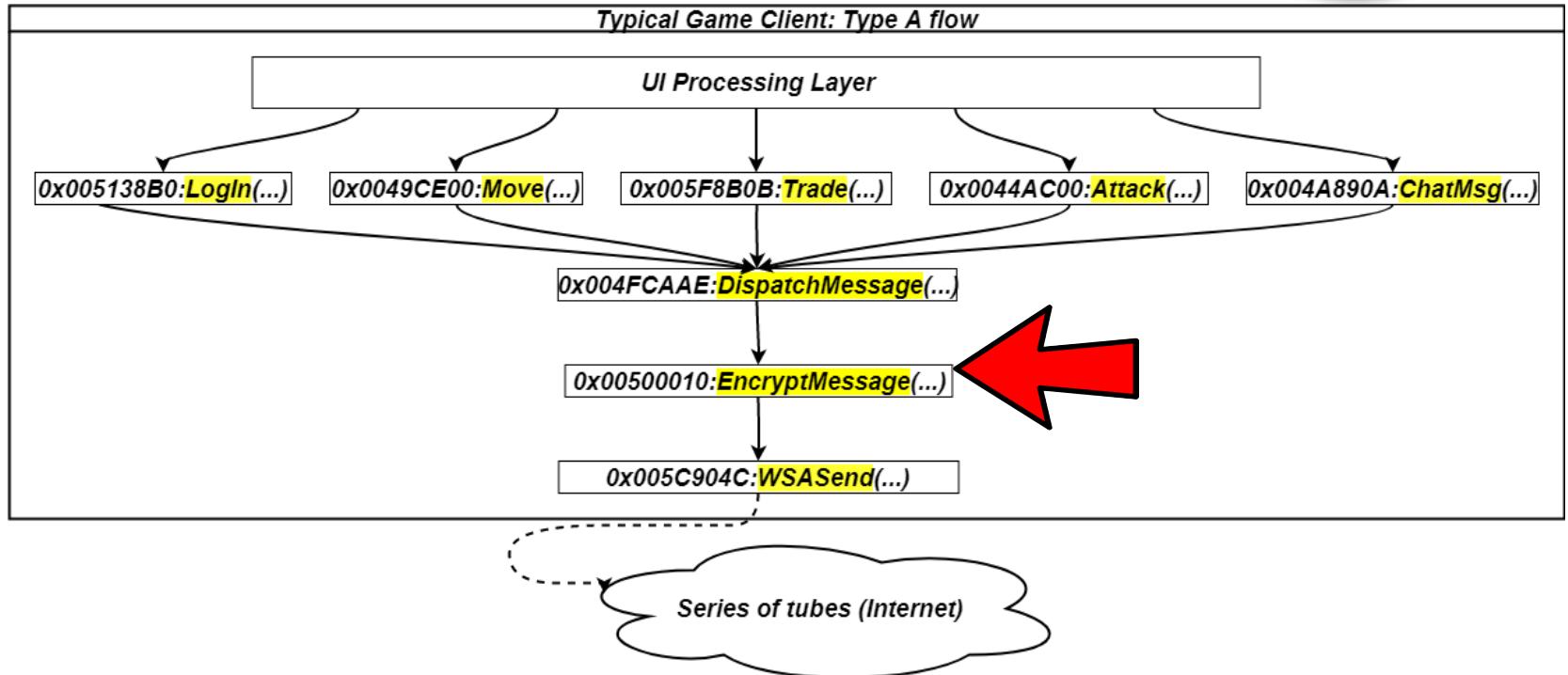
Type A Game Client (Easy)



Type A Insertion Point



Type A Insertion Point



Attack Plan – Network MITM?



Pros

- Complete takeover of the incoming and outgoing client to server messages
- Data can be routed to off computer tools to analyze and modify packets.
- Cheat/Hack detection irrelevant

Cons

- Time consuming
- Must RE encryption/Negotiation
- Must RE any hashing/checksums
- Must RE and takeover any sequencing of packets
- Must have prior knowledge of message encoding structure i.e. how do you know where data for one packet begins and ends?
- Encryption/Packet structure probably will change every patch

Attack Plan – Modify Game Client?



Pros

- Modify global values that dictate gameplay behavior
 - Movement Speed
 - Character Location
 - Wall hacking (disable collision)

Cons

- Time consuming
 - RE each behavior and find where code handling it is
- Can miss a lot of hidden code features or messages being sent to server that may do more interesting things!

Attack Plan – Application Layer MITM?



Pros

- Complete takeover of the incoming and outgoing client to server messages
- Can be done pre encryption/checksum for outgoing
- Can be done post decryption/checksum for incoming messages
- Easily inject forged packets to outgoing or incoming data stream
- Automatic detection of packet boundaries

Cons

- Need custom tools
- Lots of RE
- Client anti-cheat/modification detection is possible*

Methodology Demo

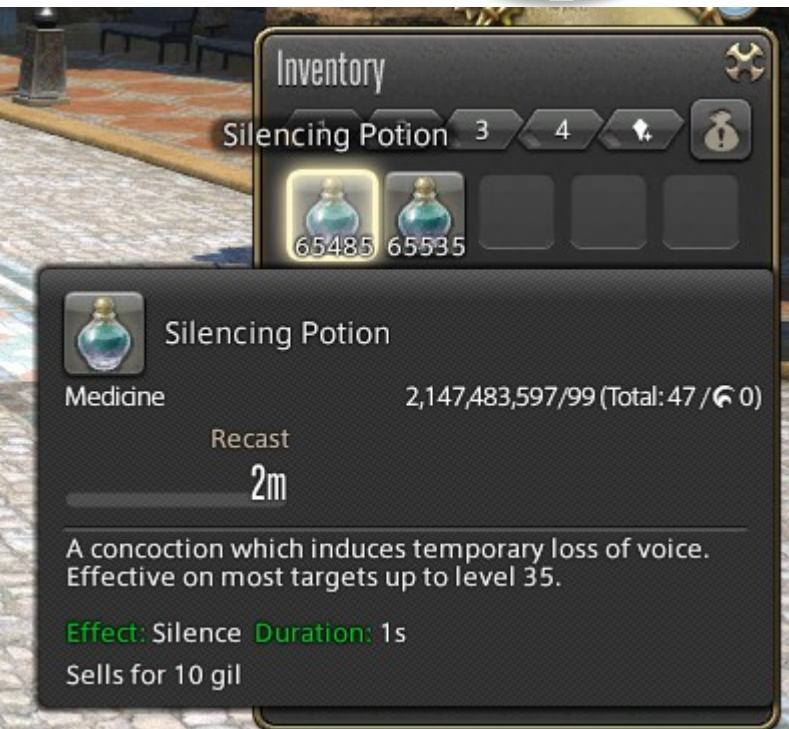


- Runtime binary data flow analysis
 - [thread-hello.mp4](#)

Summary



- Extreme obfuscation works in favor of the highly skilled adversaries
 - Low hanging fruits are protected
 - Typically 6-7 integer over-under exploits
- Expect to be probe and hacked
 - Honeypots in code, extreme logging offloading
- Take a wholistic approach to security
(Don't focus on one area)



Summary Continued.

- Security through obscurity makes it harder to discover successful attacks
- Monitor high level characteristics
 - Look for outliers in statistical data
- Don't ignore compiler warnings!
 - They can be annoying but they do help
 - (Usually...)

| Abilities | |
|--------------|-------|
| Strength | 65535 |
| Agility | 65535 |
| Stamina | 65535 |
| Intelligence | 65535 |
| Sense | 9 |
| Psychic | 9 |
| Body Dev. | 16392 |
| Nano Pool | 3288 |

RSA® Conference 2018



#RSAC

QUESTIONS?

About tools?/Other common exploits?/Anything?

RSA® Conference 2018



THANK YOU!

Adrian Bednarek
Independent Security Evaluators
@ISEsecurity

<https://www.linkedin.com/in/adrianbksd/>

Ew! Code at 9 AM!?



```
6     signed int playerGold = 1;  
7  
8     void bidOnItem(unsigned int playerBid){  
9         if (playerGold < playerBid){  
10            printf("Failed Bid!");  
11        }  
12        else{  
13            printf("Successful Bid!");  
14            playerGold -= playerBid;  
15        }  
16    }
```

It Lives!



- It compiles!
- But wait...

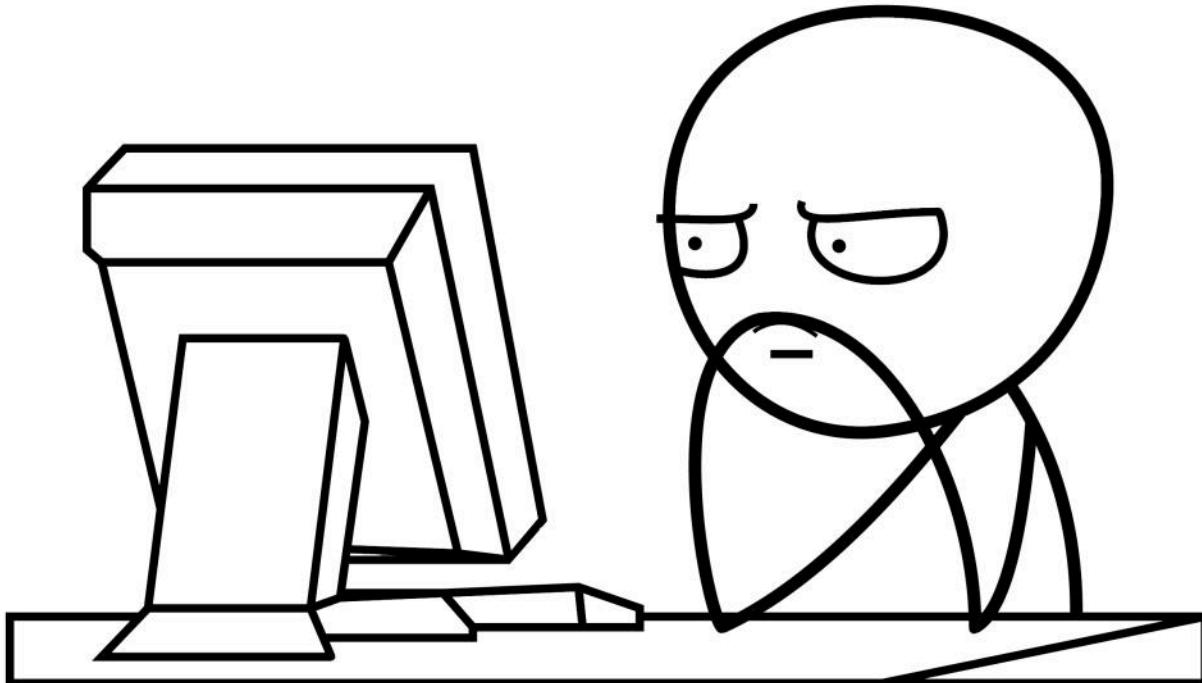
It compiles!
Let's ship it!

| Error List | | | | | |
|---------------------------------------|-------|-------------------------------|---------------|-------------------|-------------|
| Entire Solution | | 0 Errors | 1 Warning | 0 Messages | 0 |
| | Code | Description | Project | File | Line |
| ⚠ | C4018 | '<': signed/unsigned mismatch | ESOSignedness | esosignedness.cpp | 9 |
| View All Warnings | | | | | Skip to End |

Error List **Output**



I don't like warnings...compiler spam.





Easy fix!

```
6     signed int playerGold = 1;
7
8     □ void bidOnItem(unsigned int playerBid){
9         □ if (playerGold < (signed int)playerBid){
10            printf("Failed Bid!");
11        }
12        □ else{
13            printf("Successful Bid!");
14            playerGold -= playerBid;
15        }
16    }
```



Easy fix!

```
6     signed int playerGold = 1;
7
8     □ void bidOnItem(unsigned int playerBid){
9         □ if (playerGold < (signed int)playerBid){
10            printf("Failed Bid!");
11        }
12        □ else{
13            printf("Successful Bid!");
14            playerGold -= playerBid;
15        }
16    }
```

RSA® Conference 2018



COMPILED/NO WARNINGS/ALL SET/DEPLOY!

Few days/weeks later: There's some weird stuff going on. Is my server haunted?

