Game Exploitation

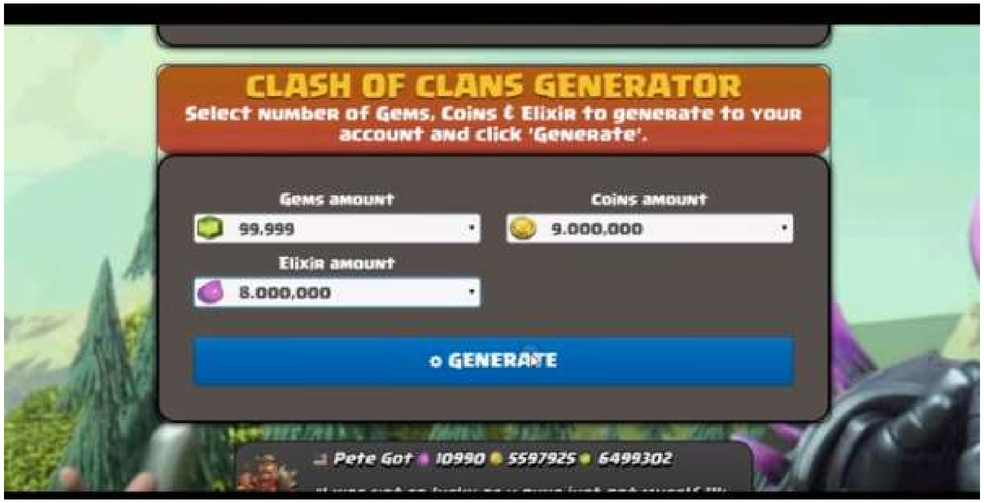
CSCI4120 Principle of
Computer Game Software

Game Exploitation

- Whether some exploits can be treated as cheats are up to debate
  - involves the argument that the issues are part of the game and require no changes or external programs to take advantage of them.
- Cheating in games – single player, online games
  - For single play, cluebook as well as cheat keys are abundant which is no longer secret
  - Eg. Assassin‘s Creed: Odyssey offers players a means of boosting their XP generation rate by 50 percent – for just a US$10

Game Exploitation

- Motivations
  - Do it for living
  - Gamers gone mad by difficulties set
  - Gamers too busy to devote more time



CLASH OF CLANS GENERATOR

Select number of Gems, Coins & Elixir to generate to your account and click 'Generate'.

Gems amount: 99.999

Coins amount: 9.000.000

Elixir amount: 8.000.000

Pete Got 10990 5597925 6499302

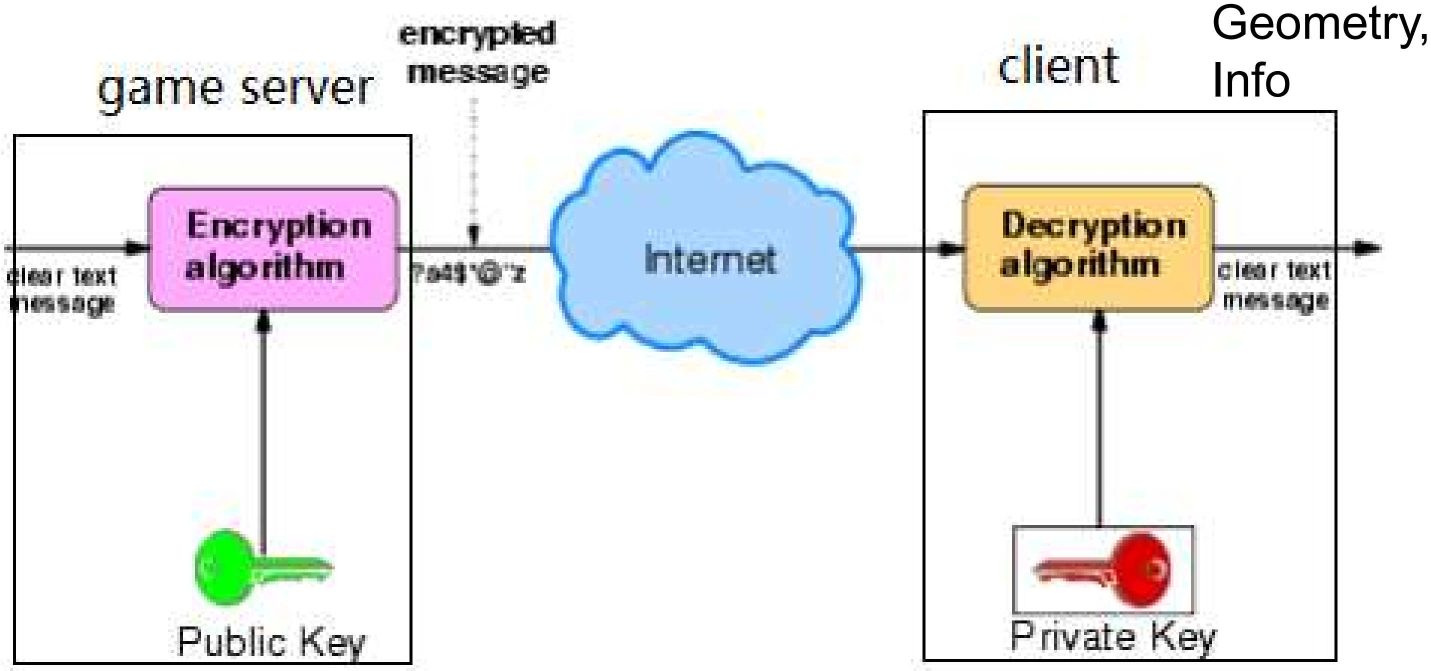
Gems hack in CoC : unlimited resources Hack (scam only)

Game Exploitation

- As developers, game companies try every effort to stop exploits as it would hurt the player community
- Usual measures are setting up policy together with effective countermeasure to flag out exploit players
- Less than a month into 2017, Blizzard has already banned more than 10,000 *Overwatch* South Korean accounts
- Punishment includes sanctions against teams and players, even disqualification

Game Exploitation

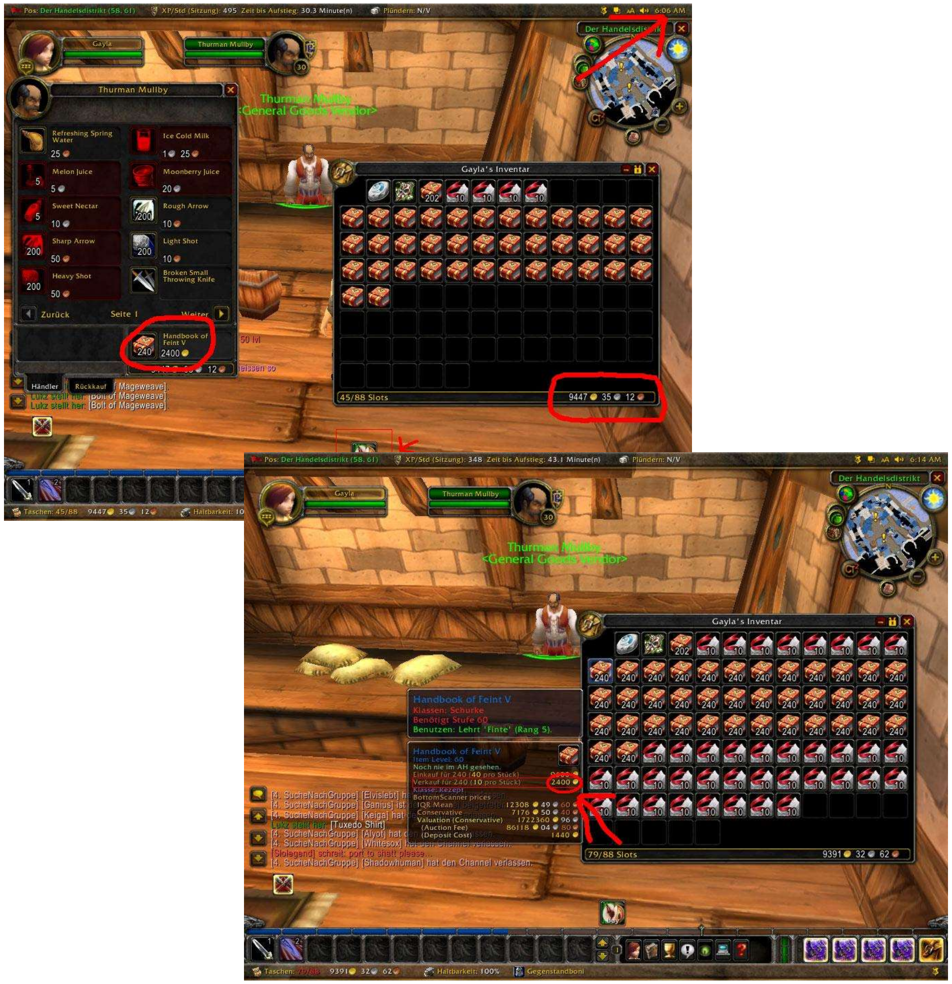
- Online games are the biggest target
- First battlefield is the game client
- “The client is in the hands of the enemy.” — *The Laws of Online World Design*



Geometry, game play Info

Common Exploits

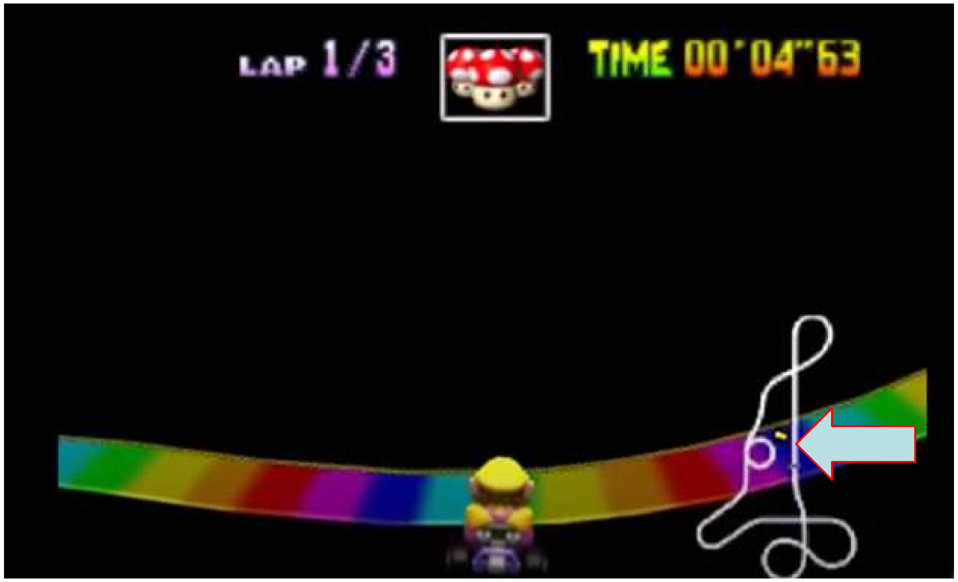
- Duping
  - duplicating items or money
- Lag / disconnection
  - Games with inadequate lag handling may let players intentionally cause lag themselves to cause an advantage
  - Game that let players disconnect immediately without consequences also let player exit without loss



Item duping in WoW

Common Exploits

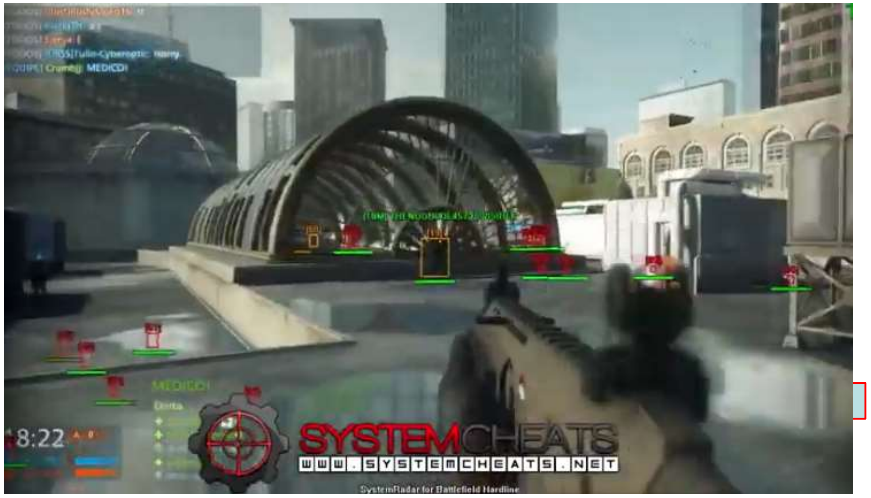
- Geometry
  - Reach normally inaccessible areas or take shortcut
  - Achieve by going through walls, etc.



Shortcut in Rainbow Road in Mario Kart 64

Common Exploits

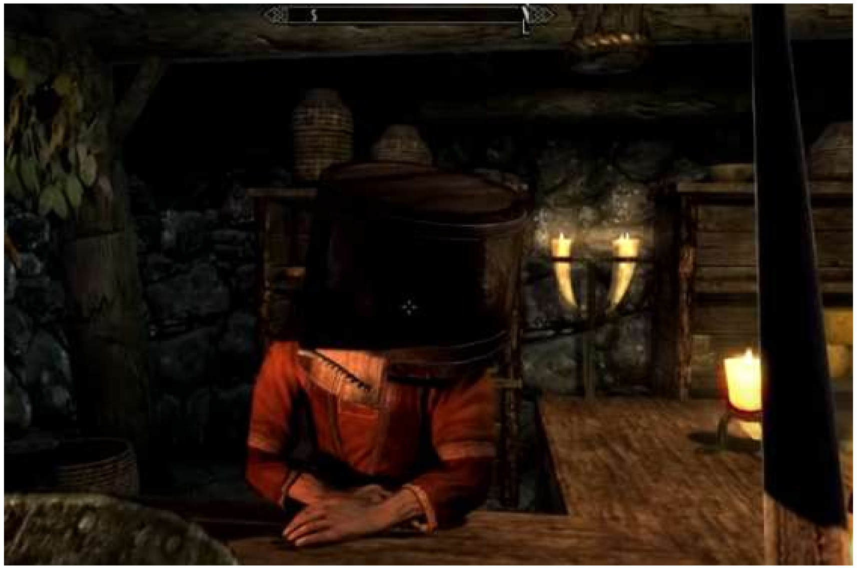
- Data stream manipulation
  - involves a player altering the flow of information between the game server and clients
  - Usually called bot



Aimbot in FPS

Common Exploits

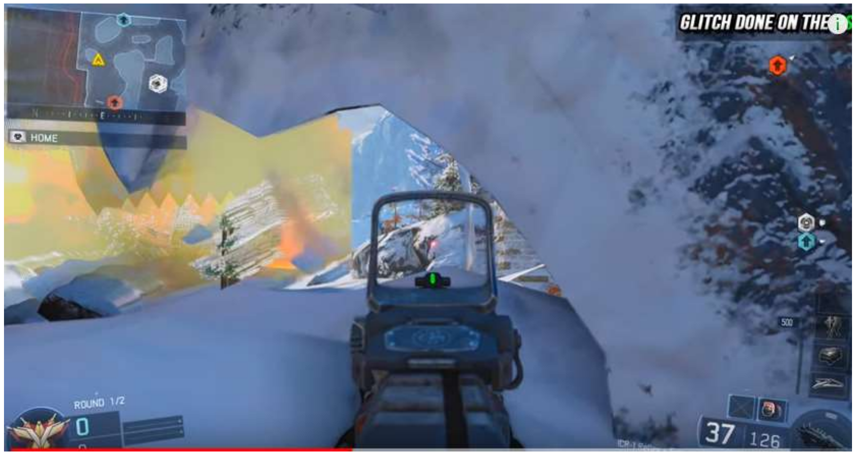
- Game mechanics
  - Take advantage of systems that make up the game play
  - Not a bug – working as designed, but is not working as intended e.g. Incorrectly balanced skill tree in online RPG game



Bucket head in Skyrim to cover shopkeeper
With bucket and steal everything

Common Exploits

- Safe zones
  - Places where a player can attack with no risk of being attacked back
  - https://www.youtube.com/watch?v=wh5Ve_pdGZI



Rock glitches in CoD : BO3 enemies can't hurt you, but you can fire at them

Common Exploits

- Movement
  - Speed hack : allow the player to move faster than intended
  - Wall hack : usually do client-side collision check. Disable it to explore all static data set client has loaded

Speed run in Super Mario 64

Details and Countermeasure

- Bot Building
- Taking advantage of bugs
- Hacking the client
- Finding the future

Building a Bot

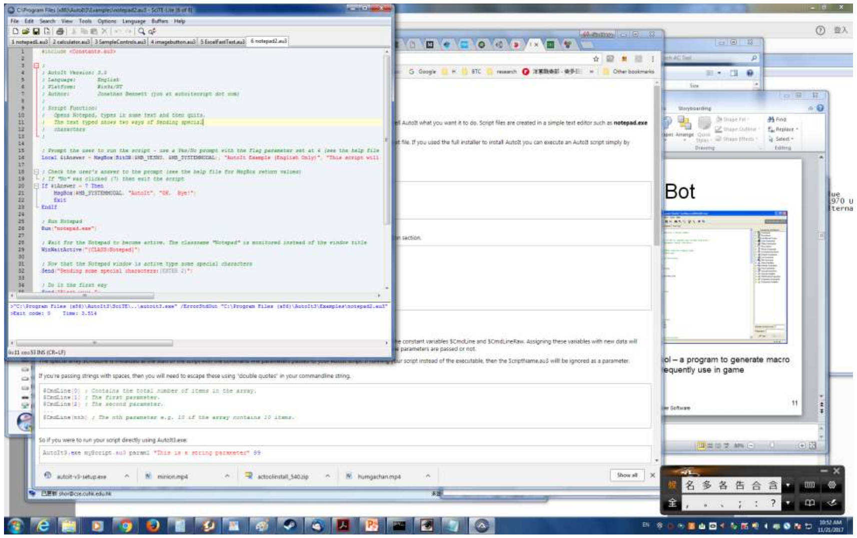
- Bot (外掛)
  - standalone program that play a game for you
  - Keyboard mapping program that script several common actions or AI scripts
  - Common in FPS for perfect aiming
  - Online poker bot can win on basic tables with some regularity

- Blizzard has banned 320,000 *Warcraft III* and *Diablo2* accounts in 2010, mainly for use of 3rd party program i.e. bots

Building a Bot

- Using User Interface
  - Many games has UI elements on screen
  - Using special tools/scripts (macro) to automate game play

- Example
  - World of Warcraft(WoW) monsters appear at specific location on a periodic basis
  - You can write a macro/bot waiting at that location to automate the repetitive experience increasing activity



```
#include <Constants.au3>

;
; AutoIt Version: 3.0
; Language:       English
; Platform:       Win9x/NT
; Author:         Jonathan Bennett (jon at autoitscript dot com)
;
; Script Function:
;   Opens Notepad, types in some text and then quits.
;   The text typed shows two ways of Sending special
;   characters
;

; Prompt the user to run the script - use a Yes/No prompt with the flag parameter set at 4 (see the help file
Local $iAnswer = MsgBox(BitOR($MB_YESNO, $MB_SYSTEMMODAL), "AutoIt Example (English Only)", "This script will

; Check the user's answer to the prompt (see the help file for MsgBox return values)
; If "No" was clicked (7) then exit the script
If $iAnswer = 7 Then
    MsgBox($MB_SYSTEMMODAL, "AutoIt", "OK.  Bye!")
    Exit
EndIf

; Run Notepad
Run("notepad.exe")

; Wait for the Notepad to become active. The classname "Notepad" is monitored instead of the window title
WinWaitActive("[CLASS:Notepad]")

; Now that the Notepad window is active type some special characters
Send("Sending some special characters:{ENTER 2}")

; Do it the first way
```

```
>"C:\Program Files (x86)\AutoIt3\SciTE\..\autoit3.exe" /ErrorStdOut "C:\Program Files (x86)\AutoIt3\Examples\notepad2.au3"
>Exit code: 0    Time: 3.514
```

If you're passing strings with spaces, then you will need to escape these using "double quotes" in your commandline string.

```
$CmdLine[0] ; Contains the total number of items in the array.
$CmdLine[1] ; The first parameter.
$CmdLine[2] ; The second parameter.
...
$CmdLine[nth] ; The nth parameter e.g. 10 if the array contains 10 items.
```

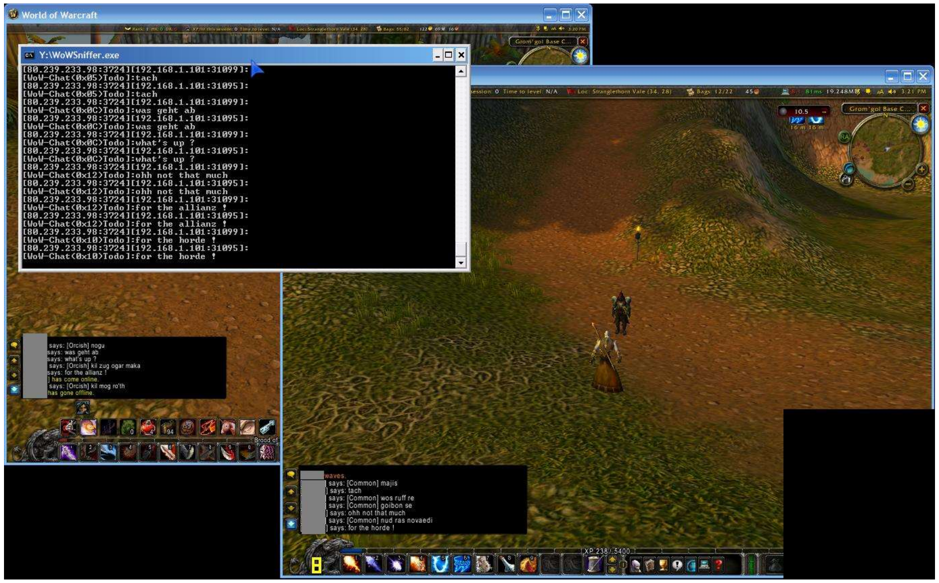
So if you were to run your script directly using AutoIt3.exe:

```
AutoIt3.exe myScript.au3 param1 "This is a string parameter" 99
```

AutoIt – a program to GUI automation
And frequently use in game

Building a Bot

- Operating a Proxy
  - Proxy acts between game client and server to intercept and alter packets
  - Monitoring the network traffic, hooking between program and dynamic link library all are methods used by proxy
- Example
  - In Counter-Strike, proxy cheats have been used to improve aim drastically
- Countermeasure
  - Encryption can solve the problem, but with extra CPU cycles needed



```
[80.239.233.98:3724][192.168.1.101:31099]:
[WoW-Chat(0x05)Todo]:tach
[80.239.233.98:3724][192.168.1.101:31095]:
[WoW-Chat(0x05)Todo]:tach
[80.239.233.98:3724][192.168.1.101:31099]:
[WoW-Chat(0x0C)Todo]:was geht ab
[80.239.233.98:3724][192.168.1.101:31095]:
[WoW-Chat(0x0C)Todo]:was geht ab
[80.239.233.98:3724][192.168.1.101:31099]:
[WoW-Chat(0x0C)Todo]:what's up ?
[80.239.233.98:3724][192.168.1.101:31095]:
[WoW-Chat(0x0C)Todo]:what's up ?
[80.239.233.98:3724][192.168.1.101:31099]:
[WoW-Chat(0x12)Todo]:ohh not that much
[80.239.233.98:3724][192.168.1.101:31095]:
[WoW-Chat(0x12)Todo]:ohh not that much
[80.239.233.98:3724][192.168.1.101:31099]:
[WoW-Chat(0x12)Todo]:for the allianz !
[80.239.233.98:3724][192.168.1.101:31095]:
[WoW-Chat(0x12)Todo]:for the allianz !
[80.239.233.98:3724][192.168.1.101:31099]:
[WoW-Chat(0x10)Todo]:for the horde !
[80.239.233.98:3724][192.168.1.101:31095]:
[WoW-Chat(0x10)Todo]:for the horde !
```

[redacted]
```
 says: [Orcish] nogu
says: was geht ab
says: what's up ?
 says: [Orcish] kil zug ogar maka
says: for the allianz !
] has come online.
 says: [Orcish] kil mog ro'th
has gone offline.
```

[redacted]
```
waves.
] says: [Common] majis
] says: tach
] says: [Common] wos ruff re
] says: [Common] goibon se
] says: ohh not that much
] says: [Common] nud ras novaedi
] says: for the horde !
```

WoWSniffer : a program to intercept communication between WoW client and server

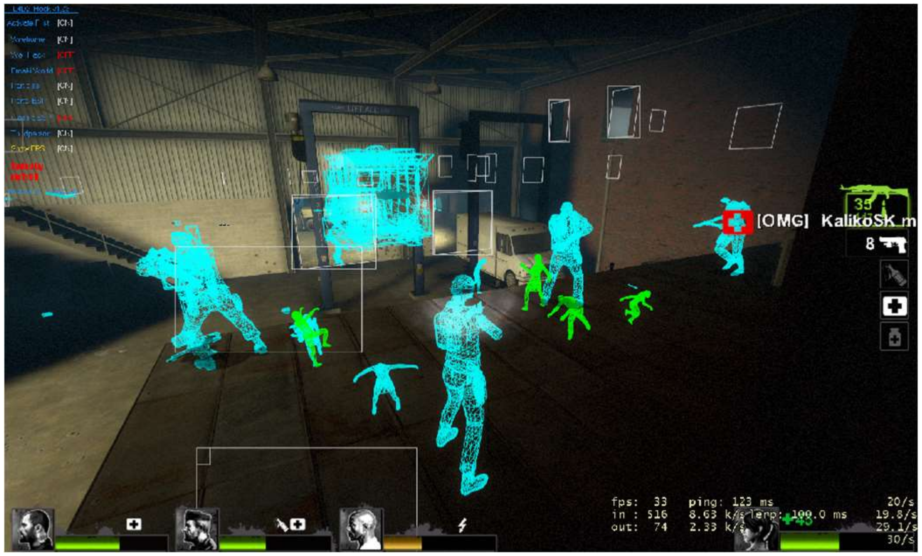
Hacking the Client

- Manipulating Memory
  - reading and writing memory (game states) directly
  - Common ways involve the graphics driver
    - In FPS, the OpenGL or Direct3D driver is the primary target
Reasons:
MS Windows didn't have a way to identify/prevent the driver is legitimate or not.

Aimbot in PUBG

Hacking the Client

- Manipulating Memory
  - Packet altering : change packet before get interpreted. eg. add waypoint on quest target.
  - Representation altering: replace all walls with transparent texture to add visibility
  - Triggering : have client automatically send back response eg. Autoaim
  - Spamming the server : controlling upstream for more actions
- **Just cannot trust the client**



Left 4 Dead 2 visibility hack

Finding the Future

- Many games involve chances e.g. poker
- Example – pseudo random number generator
- In 1999, a security flaw was discovered in a poker game
  - The exploit allowed a player to calculate the exact deck being used for each hand in real time
  - That means a cheater can know when to hold and fold
- The flaw exists in shuffling algorithm to generate each deck – a call to randomize() is used to reseed the generator with the current time.
- For an integer seed, there are 4 billion seeds
- The implementation using Delphi 4, seeds with the number of milliseconds since midnight according to the system clock, this amounts to 86,400,000 milliseconds in a day
- So the seed further reduced to 86,400,000
- By synchronizing the cheat program clock with that of server, possible combinations are reduced to further 200,000
- Cheat program needs to know the first 5 cards to deduce which is the current deck by searching in 200,000 possibilities.
- Reference : Building secure software by John Viega and Gary McGraw (Addison-Wesley 2001)

Taking advantage of Bug

- Seven Main area of software security
  - Input validation & representation
    - Buffer overflow
    - cross site scripting
    - SQL injection
  - API abuse
    - Misuses of API in languages or libraries
  - Mistakes in security features
  - Time and state
  - Error handling
  - Code quality
  - Encapsulation

Timing & State

- Online game situation
  - moving state around thousands of client processes on a common server in real time => race condition
  - Together with unpredictable/uncontrollable network lag
- Online game divide the game world into geographically different area (by different servers) e.g. WoW limited users to 50,000 players per server
  - Boundaries thus exist
  - When crossing border, consistent of states will be under test

Timing & State

- A typical scenario to tackle race condition (Item duping)
  - Pick a server with most laggy instance dungeon
  - Player 1 trade an item to player 2, then kill his process after entering the laggy dungeon.
  - Player 2 keep the item and wait
  - Player 1 enter game again to check whether the item still intact

Currency duping in Star Wars Galaxies

A significant currency dupe in *Star Wars Galaxies* was found after the designers compared how much money was created versus how much money was destroyed and noted that, despite more money leaving the system than entering, they observed no shortage of money in the virtual world

State Interactions

- Bugs on character states
- Spells are popular in RPG, and typically change a character states much
- Two simultaneous spells on a single body usually leads to interesting results
  - Cast spell of attack of player’s pet and flight together, resulting in steering of flight possible (player standing on the pet)

Tools to find bugs

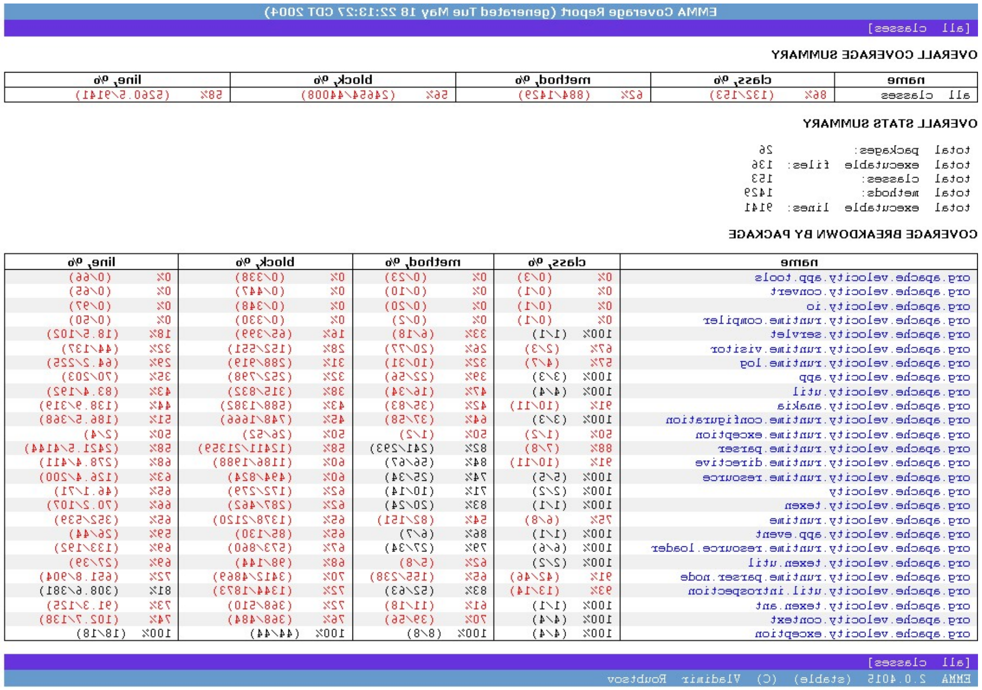
- Software testing paradigm
- Traditional ways in QA to find bugs
  - Debuggers
  - Coverage tools
  - Fault injection machine
  - Virtual machine simulation
  - Decompilers/disassemblers

Tools to find Bugs

- Decompilers
  - Yield an approximate source code from the executable
- Disassemblers
  - Translate binary to assembly code
- Typically would start from decompilers, if doesn’t work, then try disassemblers
- Debugger
  - Using kernel level debugger and setting breakpoints or triggers to stop a process in the midst
  - Triggers may look for certain messages such as secret key press or use of particular functions

Tools to find bugs

- Coverage tools
  - Determine when a given part of program is run under a specific test
- Using code scan tool, an attacker might have discovered an exploitable defect
- Using coverage tool can help to get there with just the right state data



EMMA Coverage Report (generated Tue May 18 22:13:27 CDT 2004)

[all classes]

OVERALL COVERAGE SUMMARY

name	class, %	method, %	block, %	line, %
all classes	86% (132/153)	62% (884/1429)	56% (24654/44008)	58% (5260.5/9141)

OVERALL STATS SUMMARY

total packages:	26
total executable files:	136
total classes:	153
total methods:	1429
total executable lines:	9141

COVERAGE BREAKDOWN BY PACKAGE

name	class, %	method, %	block, %	line, %
org.apache.velocity.app.tools	0% (0/3)	0% (0/23)	0% (0/338)	0% (0/66)
org.apache.velocity.convert	0% (0/1)	0% (0/10)	0% (0/447)	0% (0/65)
org.apache.velocity.io	0% (0/1)	0% (0/20)	0% (0/348)	0% (0/97)
org.apache.velocity.runtime.compiler	0% (0/1)	0% (0/2)	0% (0/330)	0% (0/50)
org.apache.velocity.servlet	100% (1/1)	33% (6/18)	16% (65/399)	18% (18.5/102)
org.apache.velocity.runtime.visitor	67% (2/3)	26% (20/77)	28% (152/551)	32% (44/137)
org.apache.velocity.runtime.log	57% (4/7)	32% (10/31)	31% (288/919)	29% (64.2/225)
org.apache.velocity.app	100% (3/3)	39% (22/56)	32% (252/798)	35% (70/203)
org.apache.velocity.util	100% (4/4)	47% (16/34)	38% (315/832)	43% (83.4/192)
org.apache.velocity.anakia	91% (10/11)	42% (35/83)	43% (588/1382)	44% (138.9/319)
org.apache.velocity.runtime.configuration	100% (3/3)	64% (37/58)	45% (748/1666)	51% (186.5/368)
org.apache.velocity.runtime.exception	50% (1/2)	50% (1/2)	50% (26/52)	50% (2/4)
org.apache.velocity.runtime.parser	88% (7/8)	82% (241/293)	58% (12411/21359)	58% (2421.5/4144)
org.apache.velocity.runtime.directive	91% (10/11)	84% (56/67)	60% (1186/1988)	68% (278.4/411)
org.apache.velocity.runtime.resource	100% (5/5)	74% (25/34)	60% (494/824)	63% (126.4/200)
org.apache.velocity	100% (2/2)	71% (10/14)	62% (172/279)	65% (46.1/71)
org.apache.velocity.texen	100% (1/1)	83% (20/24)	62% (287/462)	66% (70.2/107)
org.apache.velocity.runtime	75% (6/8)	54% (82/151)	65% (1378/2120)	65% (352/539)
org.apache.velocity.app.event	100% (1/1)	86% (6/7)	65% (85/130)	59% (26/44)
org.apache.velocity.runtime.resource.loader	100% (6/6)	79% (27/34)	67% (573/860)	69% (133/192)
org.apache.velocity.texen.util	100% (2/2)	62% (5/8)	68% (98/144)	69% (27/39)
org.apache.velocity.runtime.parser.node	91% (42/46)	65% (155/238)	70% (3412/4869)	72% (651.8/904)
org.apache.velocity.util.introspection	93% (13/14)	83% (52/63)	72% (1344/1873)	81% (308.6/381)
org.apache.velocity.texen.ant	100% (1/1)	61% (11/18)	72% (368/510)	73% (91.3/125)
org.apache.velocity.context	100% (4/4)	70% (39/56)	76% (368/484)	74% (102.7/138)
org.apache.velocity.exception	100% (4/4)	100% (8/8)	100% (44/44)	100% (18/18)

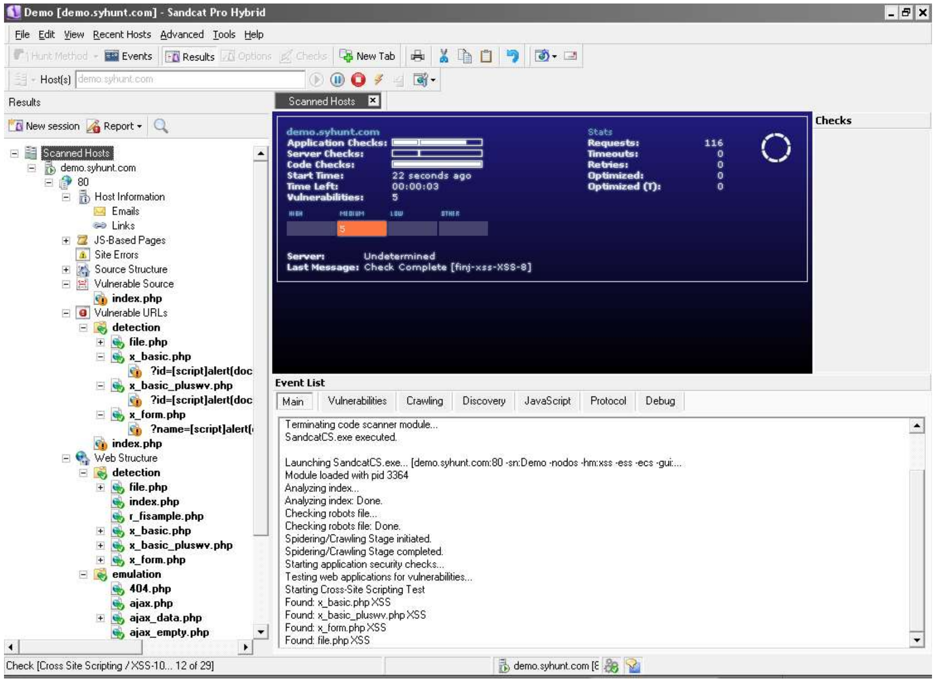
[all classes]

EMMA 2.0.4015 (stable) (C) Vladimir Roubtsov

Report on code coverage by EMMA

Tools to find bugs

- Fault injection engines
  - Perturb program and check what happens
- Consists of
  - Injection engine
  - Monitoring system
  - A way to run the program
- Typically involves generating huge sets of data to test the program



```
demo.syhunt.com                                   Stats
Application Checks:                               Requests:        116
Server Checks:                                    Timeouts:        0
Code Checks:                                      Retries:         0
Start Time:          22 seconds ago               Optimized:       0
Time Left:           00:00:03                     Optimized (T):   0
Vulnerabilities:     5

Server:        Undetermined
Last Message:  Check Complete [finj-xss-XSS-8]
```

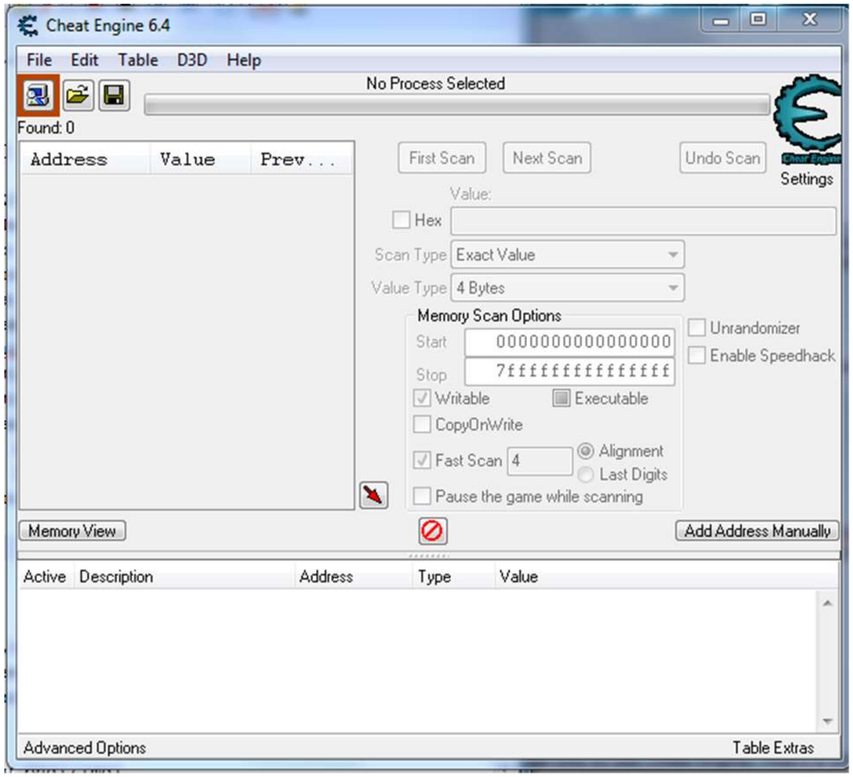
```
Terminating code scanner module...
SandcatCS.exe executed.

Launching SandcatCS.exe... [demo.syhunt.com:80 -sn:Demo -nodos -hm:xss -ess -ecs -gui:...
Module loaded with pid 3364
Analyzing index...
Analyzing index: Done.
Checking robots file...
Checking robots file: Done.
Spidering/Crawling Stage initiated.
Spidering/Crawling Stage completed.
Starting application security checks...
Testing web applications for vulnerabilities...
Starting Cross-Site Scripting Test
Found: x_basic.php XSS
Found: x_basic_pluswv.php XSS
Found: x_form.php XSS
Found: file.php XSS
```

```
Check [Cross Site Scripting / XSS-10... 12 of 29]    demo.syhunt.com [8
```

Sandcat : A fault-injection testing tool for web applications

Tools to find bugs

- Virtual machine simulators
  - Emulate a machine on top of another architecture
- Can execute software in an extremely monitored fashion
  - Rewind states
  - Execute an extreme conditions
  - Reset quickly on crash



□ Hex

Scan Type: Exact Value

Value Type: 4 Bytes

Start: 0000000000000000

Stop: 7fffffffffffffff

☑ Writable ▨ Executable

□ CopyOnWrite

☑ Fast Scan 4 ● Alignment ○ Last Digits

□ Pause the game while scanning

□ Unrandomizer

□ Enable Speedhack

Cheat Engine

Drops and Respawns

- Monsters will drop items when die
  - Special item (weapons, armors etc.) will probabilistically be dropped
- Client program will already know this (keep inventory list)
- Cheaters can pick and choose targets with special items
- In addition, when a resource/monster is removed, system will respawn, say
  - SELECT * FROM mob_instances WHERE alive=0 AND killtime>(NOW – respawn_rate)
- If respawn rate very low, then the speed of respawn for that monster/item will be higher
- Facilitate experience gathering or drop farming

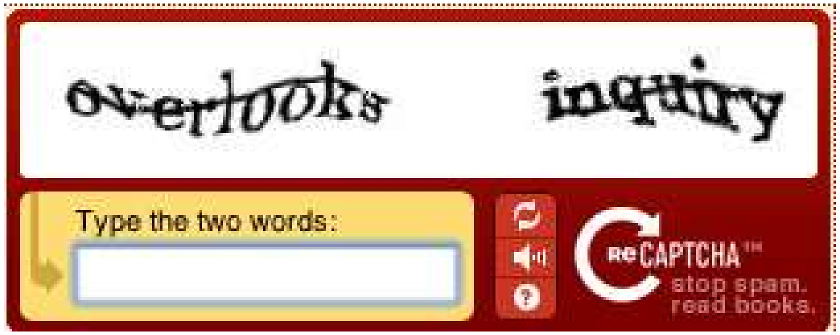
Countermeasures

- In principle, it is almost impossible to prevent a client from cheating
- To prevent code or data being modified on client side, MD5 hash checking regularly on all assets seem feasible
- But even the hash returned to server may be already tampered
- Common consensus is that reducing the client to only display and accept input (very thin client), and no more

Countermeasures

- Game companies wants to keep bots out of game

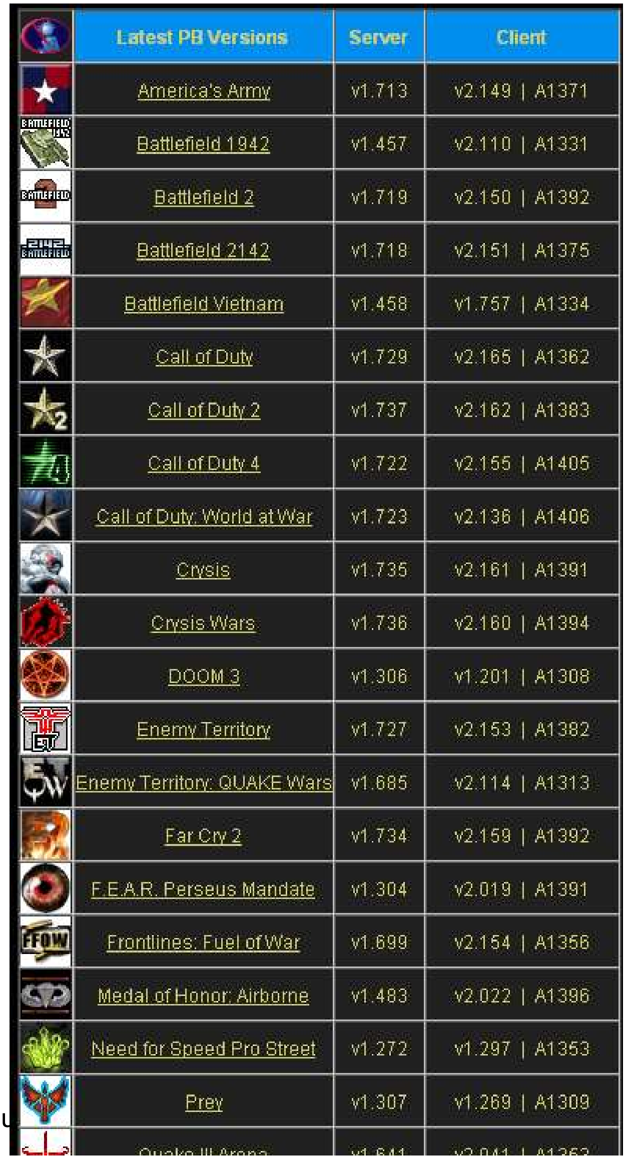
1. Reverse Turing test
   - Asking questions to ensure answering target is a human
2. Gamers self-police, try to find out bots among themselves – game chat feature



CAPTCHA – Completely Automated Public Turing Test to Tell Computers and Humans Apart

Countermeasures

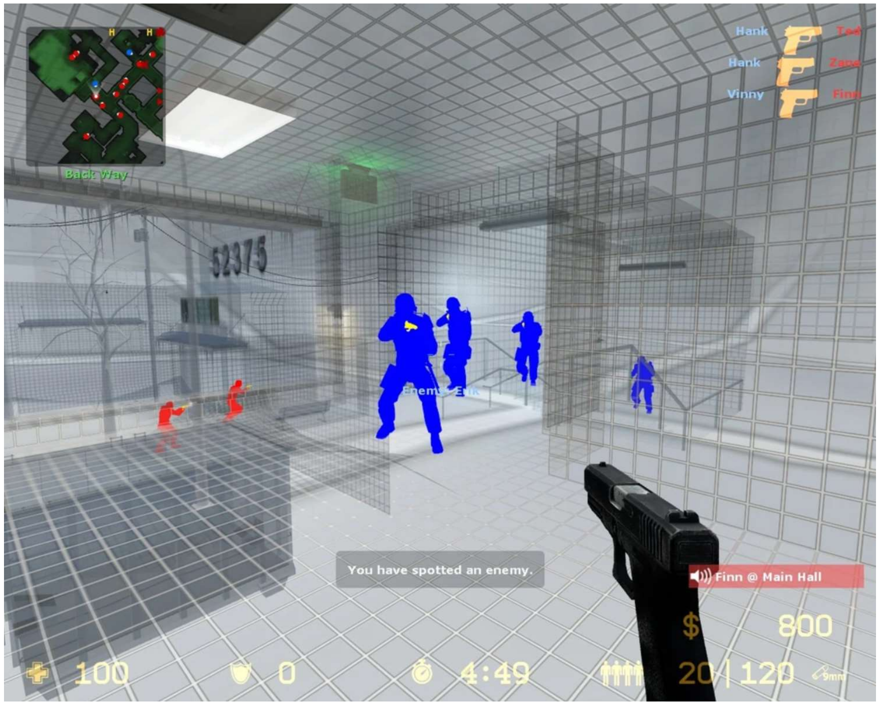
- Spyware – anti-cheat tools
- Punkbuster, which install its client on players' computers, has the following 'features'
  - Real-time scanning of memory on players' computers searching for cheats/hacks
  - Frequent status report sent to server by all players. Server raises a violation when necessary causes offending player be removed from game
  - Server randomly check player settings looking for known exploits of the game engine



Latest PB Versions	Server	Client
America's Army	v1.713	v2.149 | A1371
Battlefield 1942	v1.457	v2.110 | A1331
Battlefield 2	v1.719	v2.150 | A1392
Battlefield 2142	v1.718	v2.151 | A1375
Battlefield Vietnam	v1.458	v1.757 | A1334
Call of Duty	v1.729	v2.165 | A1362
Call of Duty 2	v1.737	v2.162 | A1383
Call of Duty 4	v1.722	v2.155 | A1405
Call of Duty: World at War	v1.723	v2.136 | A1406
Crysis	v1.735	v2.161 | A1391
Crysis Wars	v1.736	v2.160 | A1394
DOOM 3	v1.306	v1.201 | A1308
Enemy Territory	v1.727	v2.153 | A1382
Enemy Territory: QUAKE Wars	v1.685	v2.114 | A1313
Far Cry 2	v1.734	v2.159 | A1392
F.E.A.R. Perseus Mandate	v1.304	v2.019 | A1391
Frontlines: Fuel of War	v1.699	v2.154 | A1356
Medal of Honor: Airborne	v1.483	v2.022 | A1396
Need for Speed Pro Street	v1.272	v1.297 | A1353
Prey	v1.307	v1.269 | A1309
Quake III Arena	v1.641	v2.041 | A1363

Countermeasures

- Valve Anti-Cheat (VAC) is a server side anti-cheat software
- Make rules against cheating and ban violated accounts
- scanning every player that attempts to connect to a Steam server
- examines a player's actions and tests them against multiple statistical markers

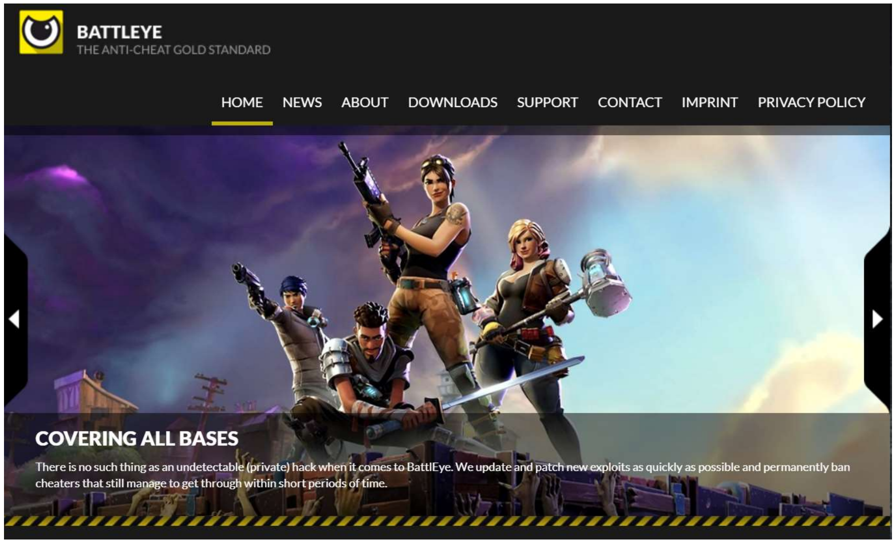


You have spotted an enemy.

A material hack, a type of content hack that said once VAC cannot detect

Countermeasures

- Against hacking Client
- Packing
  - changes the way binary executable stored on disks
  - Preventing decompiler/disassemblers
- But still defeated by unpacker, single step debugger, VM etc.



COVERING ALL BASES

There is no such thing as an undetectable (private) hack when it comes to BattlEye. We update and patch new exploits as quickly as possible and permanently ban cheaters that still manage to get through within short periods of time.

BattleEye, a popular anti-cheat service

Countermeasures

- Against hacking Client
- Anti-debugging
  - Check system to see if debugger is present
  - Defeated by clearing the BeingDebugged flag in the process environment block(PEB)
- Forwarding exception
  - Game program set up exception handler that is supposed to be called when some on purpose bugs occur
  - Then throw the on-purpose bug
  - Game program then check whether the handler has been called
  - If no, then a debugger is attached
- Again, defeat by forwarding all exception back to game program

Countermeasures

- Against hacking Client
  - Game program read system time at normal intervals
  - Will know it is being debugged when sample grows beyond certain threshold

Countermeasures

- Against macro bot
  - Macro program is hardly hacking as it just simulate a real player by pressing keys & mouse
  - Resort to scanning processes and window names in system
  - Analyzing the player traffic pattern using machine learning [5]

References

1. *Exploiting Online Games: Cheating Massively Distributed Systems*, Greg Hoglund, Gary McGraw, Addison-Wesley, 2007
2. Exploitation Policy of Battle.net https://us.battle.net/support/en/article/exploitation-policy
3. https://en.wikipedia.org/wiki/Exploit_(video_gaming)
4. https://www.raphkoster.com/2008/04/17/how-to-hack-an-mmo/
5. https://www.utdallas.edu/news/science-technology/stopping-video-game-cheaters-2020/?WT.mc_id=NewsHomePageCenterColumn