iSECpartners
part of nccgroup

Next Level Cheating and Leveling Up Mitigations

Nicolas Guigo **Joel St. John**

Agenda

iSECpartners part of nccgroup

- A brief history of cheating in video games
- Current state of the arms race (cheating vs anti-cheat)
- The future of cheating
- Attacking anti-cheat software
- Solutions and conclusions

iSECpartners
part of nccgroup

The Money Aspect

- Multi-billion dollar industry
- Subscription models
- Streaming/Sponsorship
- Virtual grey market

iSECpartners
part of nccgroup

What is Cheating?

- Unfair advantage
  - Abusing game logic
    - Multi-accounts
    - Botting/Scripting
    - Manipulating extraneous client-side data
  - Exploiting client / server code bugs
    - Abusing bugs/glitches
    - Attacking other players or the game server

iSECpartners
part of nccgroup

A History of Cheating

- Early computer games
- Early multiplayer games
- Modern multiplayer games
- Examples!

iSECpartners
part of nccgroup

Common Cheating Vectors

- Speed/Movement hacks
- Botting
- Scripting

iSECpartners
part of nccgroup

Common Cheating Vectors

- Speed/Movement hacks
- Botting
- Scripting
- Player/item finding hacks


Charcoal Tabs
31
Baseball Bat
19
Rotten Orange
33
7.62mm Roundse Wrench
35
10
7.62mm 20 Rounds
Rotten Banana
12
Green Pen
Zluta Kolalbka Soda
20
19
Blue Pen
17
Black Hoodie
18
13
Zombie
27
T-Shirt White
27
First Aid Kit
2
Dallas mask
5

iSECpartners
part of nccgroup

Common Cheating Vectors

- Speed/Movement hacks
- Botting
- Scripting
- Player/item finding hacks
- Wall hacks/x-ray mods


23
11
2
34
4
6


-138
68
267

The Rise of Anti-Cheat

iSECpartners part of nccgroup

- Warden (~2004)
  - World of Warcraft
  - Starcraft 2
- Valve Anti-Cheat (VAC, 2002)
  - Counter-Strike
  - Team Fortress 2
- BattlEye (2004)
  - Arma 2/3
  - Day-Z

- User-land
- Reactive
- Only a mitigation

iSECpartners
part of nccgroup

The Current State of Cheating in Games

- DLL injection (internal cheating)
  - Loader
  - DLL implementing cheat logic
    - Hook Direct3D calls
    - Read/Write memory
- Network packet manipulation
  - Modify packets in-transit
  - Repeat packets
  - Introduce artificial lag
- External cheating
  - ReadProcessMemory / WriteProcessMemory
  - Transparent window

iSECpartners part of nccgroup

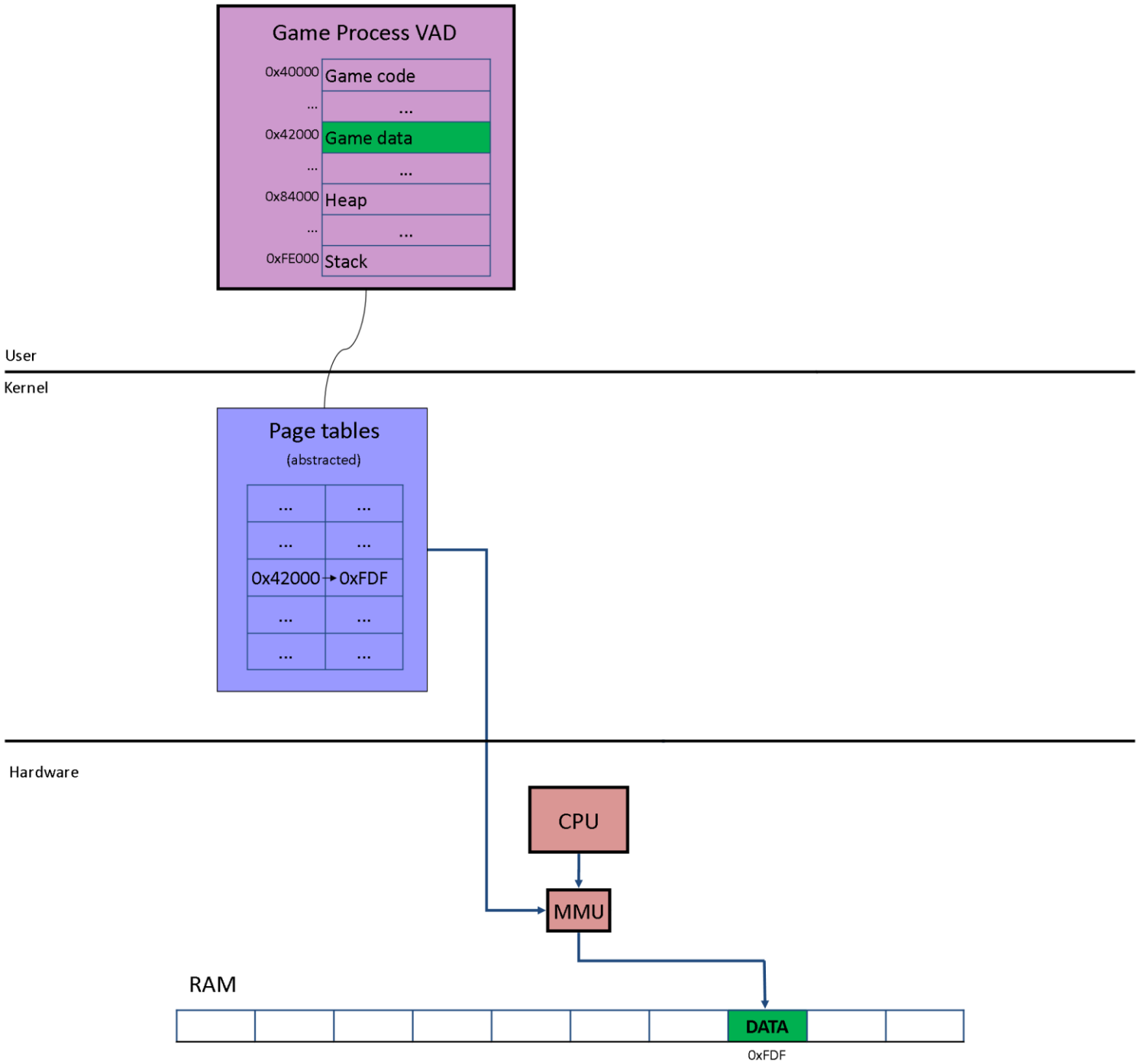
Current State of Anti-Cheat

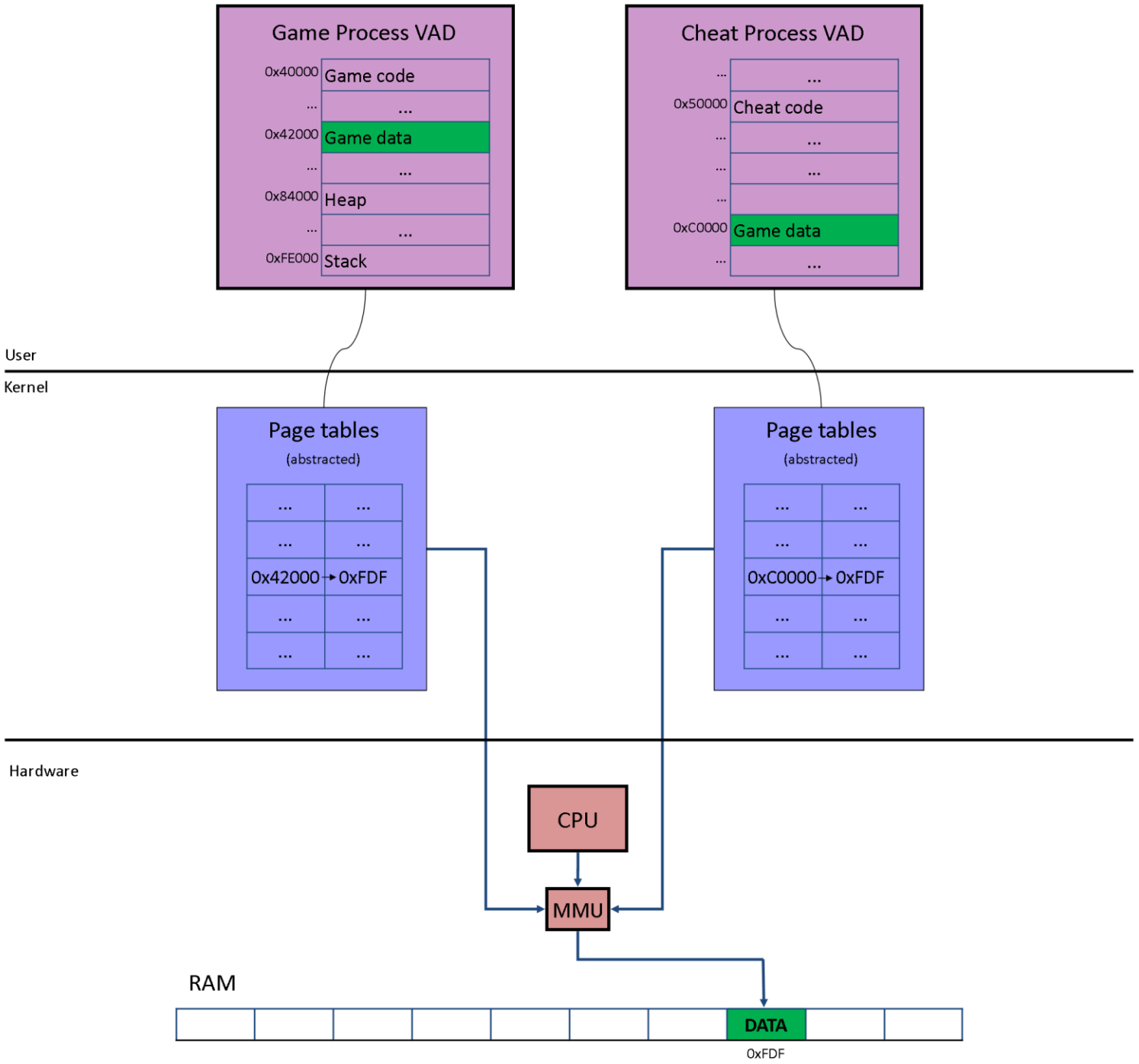
- In process
  - Signature checks
  - Game specific checks
  - Hook detection
    - Pointer chain checks
  - Call stacks periodic checks
  - Debug related detections
- Out of process
  - Signature based detection
  - Pattern searching in all processes address space
- Various
  - Scanning for game process handles
  - Scanning files for signatures (offline)
    - Send suspected programs to server for analysis
  - Check DNS history for cheat update servers
  - Etc.

iSECpartners
part of nccgroup

The Future of Cheating

- Architecture
  - Rootkit-like functionality to hide activity
  - Kernel driver
    - Makes the UM portion a protected process (DRM)
    - Maps pages from game memory into the cheat process
    - Install a filter device on the FS stack (TBD)
  - User mode executable
    - Keeps track of game/cheat mappings
    - Implements the cheat logic


Game Process VAD
0x40000
Game code
...
...
0x42000
Game data
...
...
0x84000
Heap
...
...
0xFE000
Stack
User
Kernel
Page tables
(abstracted)
...
...
0x42000
0xFDF
Hardware
CPU
MMU
RAM
DATA
0xFDF


Game Process VAD
0x40000
Game code
...
0x42000
Game data
...
0x84000
Heap
...
0xFE000
Stack
Cheat Process VAD
0x50000
Cheat code
0xC0000
Game data
User
Kernel
Page tables
(abstracted)
0x42000
0xFDF
Page tables
(abstracted)
0xC0000
0xFDF
Hardware
CPU
MMU
RAM
DATA
0xFDF

iSECpartners
part of nccgroup

Dual mapping snippet

```
status = PsLookupProcessByProcessId((PVOID)ncmmap->process, (PEPROCESS*)&epb);
if(NT_SUCCESS(status)) {
    ncmdl = NcAllocateMDL((PVOID)ncmmap->baseAddress, ncmmap->len);
    if(ncmdl) {
        KeStackAttachProcess(epb, &kapcstate);
        MmInitializeMdl(&ncmdl->mdl, (PVOID)ncmmap->baseAddress, (SIZE_T)ncmmap->len);
        _try {
            MmProbeAndLockPages(&ncmdl->mdl, UserMode, IoWriteAccess);
        }
        _except(EXCEPTION_EXECUTE_HANDLER) {
            getout=TRUE;
        }
        KeUnstackDetachProcess(&kapcstate);
        if(!getout) {
            _try {
                userva = (DWORD64)MmMapLockedPagesSpecifyCache(&ncmdl->mdl, UserMode, MmCached, NULL, FALSE, NormalPagePriority);
            }
            _except(EXCEPTION_EXECUTE_HANDLER) {
                userva = 0;
            }
            if(userva) {
                ncmdl->mdl.StartVa=(PVOID)userva;
            }
            MmUnlockPages(&ncmdl->mdl);
        } // if !getout
        else {
            ExFreePoolWithTag(ncmdl, NCDRIVER_TAG);
        }
    } // if ncmdl
    ObDereferenceObject((PVOID)epb);
} // if process
```

iSECpartners
part of nccgroup

Dual-mapping demo

iSECpartners
part of nccgroup

Pros / Cons

- Strengths
  - Generic
  - Virtually undetectable from user-mode
  - Straightforward conversion from publicly available cheat sources
  - Good performance
- Weaknesses
  - Can be challenged by KM anti-cheat
  - Run in debug mode or use signed driver

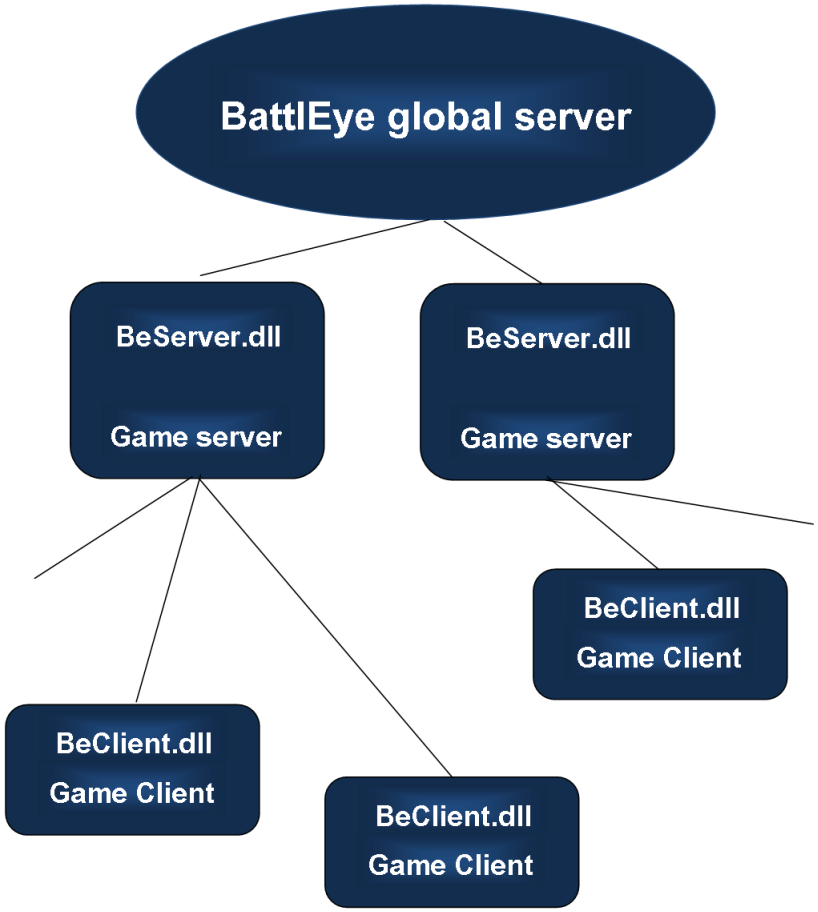
iSECpartners
part of nccgroup

Attacking Anti-Cheat Software

- Anti-cheat libraries create additional attack surface
  - On client
  - On server
- This attack surface is common to multiple games
- What happens if there is a flaw?

BattlEye

- General architecture
  - On the client
    - DLL in game process
    - System service
  - On the server
    - DLL in game server process
  - Master server
- Hooks game recv() call



iSECpartners
part of nccgroup

BE Packet structure

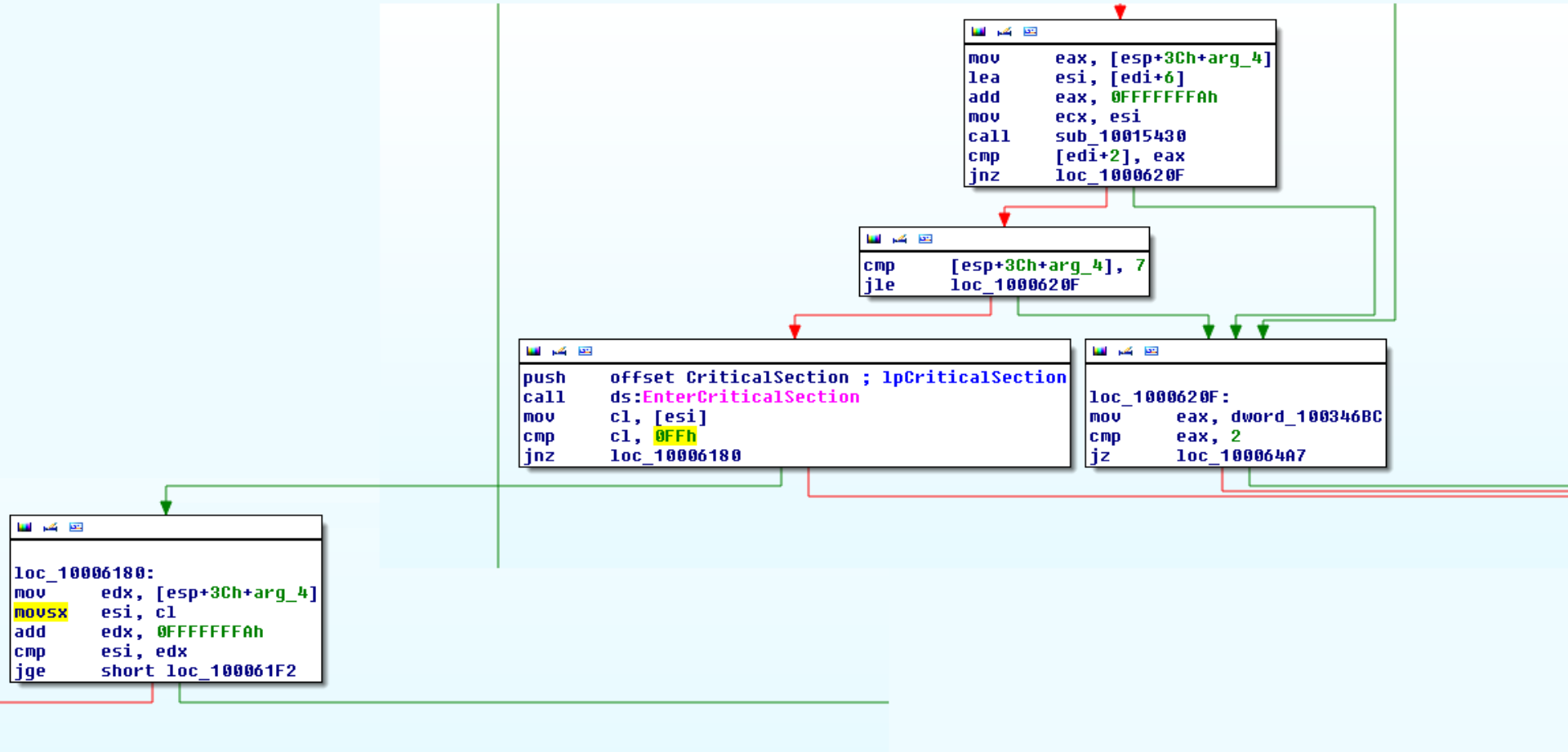
- Packet structure

'B'	'E'	H_1	H_2	H_3	H_4	LEN	DATA

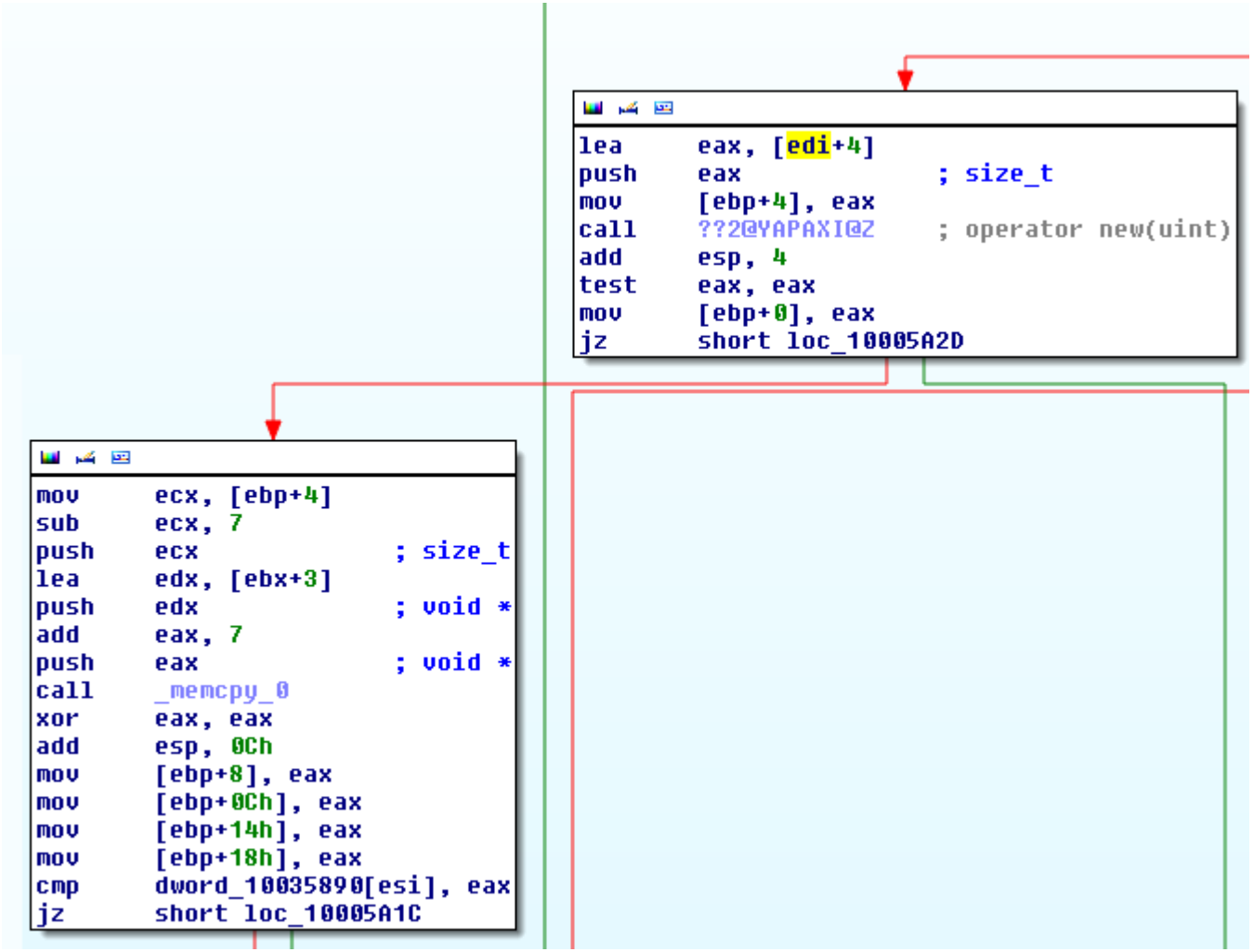
- 2 bytes signature
- Hash
- Len /code
- data

iSECpartners
part of nccgroup

Sign extension



Integer overflow -> heap overwrite

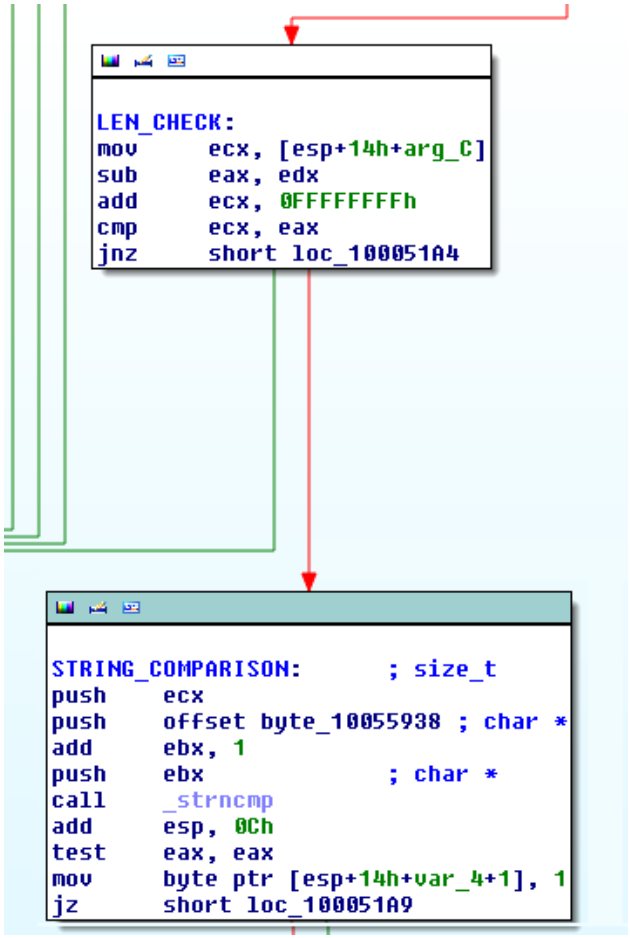


iSECpartners
part of nccgroup

Exploitability

- Denial of Service is trivial
- Remote code execution possible
  - Overwriting heap data
  - Attacker-controlled data
- Very difficult
  - Separate heap limits attack surface
    - Tool: https://github.com/iSECPartners/vtfinder
  - Race condition
    - Code execution must be achieved before thread crashes
    - Must then prevent crash from happening

BattlEye console timing attack



- Length check
- String comparison

iSECpartners
part of nccgroup

BattlEye timing attack demo

iSECpartners
part of nccgroup

Disclosure timeline

- Both vulnerabilities
  - Verified 08/2014
  - Disclosed to vendor 08/2014
  - Bugs
    - Memory corruption | fixed
    - Login vulnerability | unpatched (to date)

iSECpartners
part of nccgroup

The Future of Anti-Cheat

- Mitigations
  - Move the arms race to the kernel
  - Human factor
- Solutions
  - Full streaming of games
  - Closed platform

iSECpartners
part of nccgroup

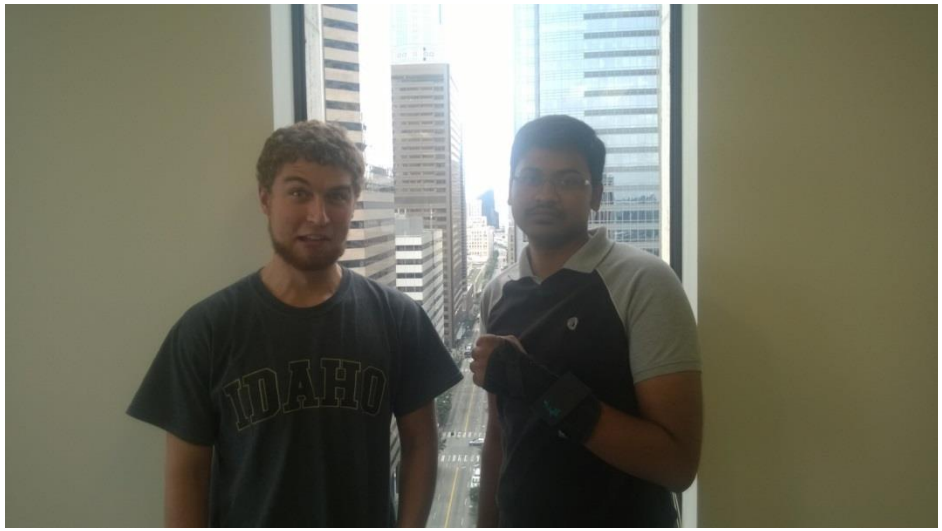
Conclusion

- Anti-cheat is a mitigation at best
- Anti-cheat creates additional attack surface
- Current anti-cheat can be completely bypassed
- Fundamental design changes are needed

Questions

iSECpartners part of nccgroup

- Thank you
  - Rachel Engel & Jason Bubolz
  - Rohit Shambhuni (iSEC 2014 Intern, Arizona State)
  - Taylor Trabun (iSEC 2014 Intern, University of Idaho)
  - Too many iSECers to list



Interns are people too!

References

- Boneh, D. and Brumley, D (2003). Remote timing attacks are practical. 12th Usenix Security Symposium. http://crypto.stanford.edu/~dabo/pubs/papers/ssl-timing.pdf
- Vtfinder. https://github.com/iSECPartners/vtfinder

iSECpartners
part of nccgroup

UK Offices
Manchester - Head Office
Cheltenham
Edinburgh
Leatherhead
London
Thame

North American Offices
San Francisco
Atlanta
New York
Seattle

Australian Offices
Sydney

European Offices
Amsterdam - Netherlands
Munich – Germany
Zurich - Switzerland