Fall Show + Tell

Announcements

- DiceCTF in < 2 weeks (Feb 4)
    - Be there!
    - Let’s be top 3
    - There will be pizza

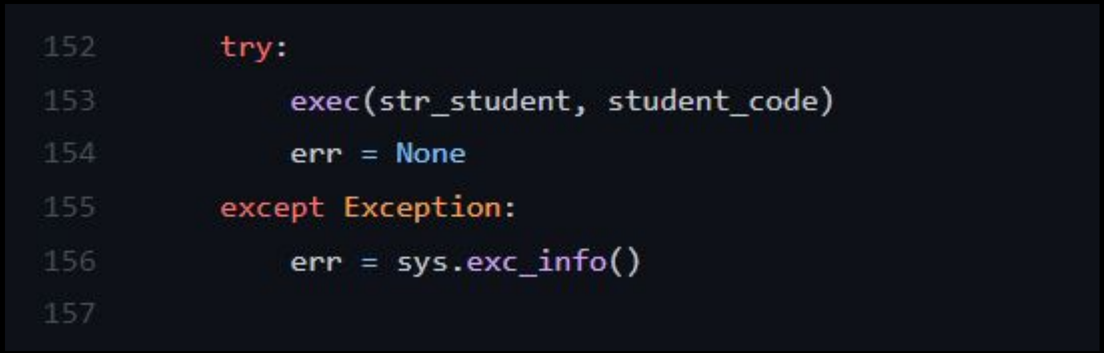
- eCTF
    - Officially started
    - Check out #eCTF on Discord

Research Projects

- Reach out to Chris if interested
- Talk to us about your ideas or if you need an idea

Pete: Breaking PrairieLearn

- Had done some work trying to sandbox same-process python execution as a discord bot in the past and failed
- Wanted to perform research on if sandboxing same-process python is possible without extreme restrictions
- While researching this, Nathan had noticed that PrairieLearn executes your code in the same process…



```
    try:
        exec(str_student, student_code)
        err = None
    except Exception:
        err = sys.exc_info()

```

Pete: Breaking PrairieLearn

- Pivoted my project to breaking out of PrairieLearn… success!
  - Command Execution on dockerized grader (I could read grader scripts, desired outputs, give myself a 100%)
- Deliverable: Wrote a patch to safely sandbox the execution (not merged yet 🤔)
- Wrote paper on my findings

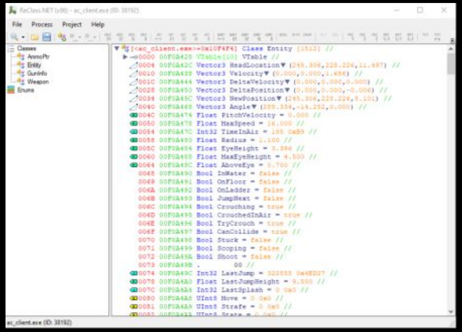
Nebu: Instruction Counting Side Channel

- Nothing very fancy: got annoyed with how irregularly Intel PIN behaves, noticed how much better Valgrind’s Callgrind is.
- PIN pain causes PinCTF to also be hard to get working.
- Wrote a “better” tool that relies on Callgrind to count instructions.
- Added some more CTF-y options.

Louis: Introductory Dive into Game Hacking

- Wanted to understand the main components of hacking a game
- C++ External vs Internal cheats
- Techniques for code injection
  - Quite often the main challenge nowadays due to kernel-level anticheats
- <u>Approach</u>: Basic open-source FPS game without anticheat
  - <u>Dynamic analysis</u>: Cheat Engine, x64dbg, ReClass.NET
  - <u>Static analysis</u>: Ghidra
  - <u>Deliverable</u>: A paper, two cheats and two injectors

```
Precision Tech AD-81 Sniper Rifle selected
A&D_Divine called a vote: load map ac_depot in mode TDM
vote passed
read map packages/maps/official/ac_depot.cgz (189 milliseconds)
The Storage Depot by MitaMAN
game mode is "TDM"
```



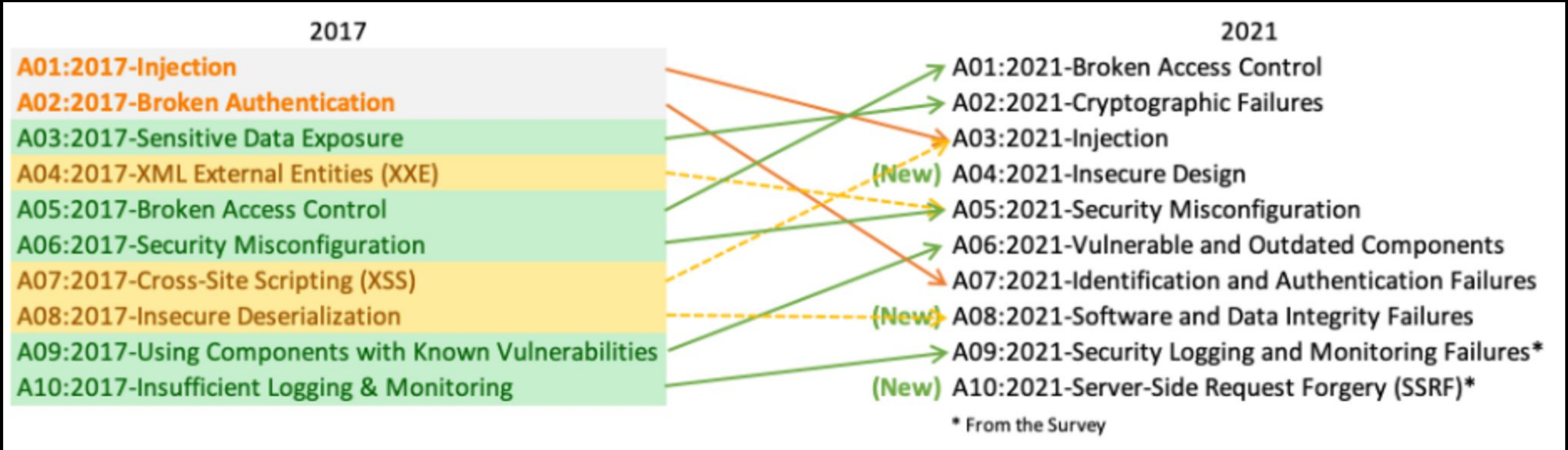
```
[<ac_client.exe>+0x10F4F4] Class Entity [1512] //
0000 00F0A428 VTable[10] VTable //
0004 00F0A42C Vector3 HeadLocation▼ (245.306,228.226,11.497) //
0010 00F0A438 Vector3 Velocity▼ (0.000,0.000,1.656) //
001C 00F0A444 Vector3 DeltaVelocity▼ (0.000,0.000,0.000) //
0028 00F0A450 Vector3 DeltaPosition▼ (0.000,0.000,-0.006) //
0034 00F0A45C Vector3 NewPosition▼ (245.306,228.226,8.101) //
0040 00F0A468 Vector3 Angle▼ (288.334,-14.252,0.000) //
004C 00F0A474 Float PitchVelocity = 0.000 //
0050 00F0A478 Float MaxSpeed = 16.000 //
0054 00F0A47C Int32 TimeInAir = 185 0xB9 //
0058 00F0A480 Float Radius = 1.100 //
005C 00F0A484 Float EyeHeight = 3.396 //
0060 00F0A488 Float MaxEyeHeight = 4.500 //
0064 00F0A48C Float AboveEye = 0.700 //
0068 00F0A490 Bool InWater = false //
0069 00F0A491 Bool OnFloor = false //
006A 00F0A492 Bool OnLadder = false //
006B 00F0A493 Bool JumpNext = false //
006C 00F0A494 Bool Crouching = true //
006D 00F0A495 Bool CrouchedInAir = true //
006E 00F0A496 Bool TryCrouch = true //
006F 00F0A497 Bool CanCollide = true //
0070 00F0A498 Bool Stuck = false //
0071 00F0A499 Bool Scoping = false //
0072 00F0A49A Bool Shoot = false //
0073 00F0A49B .         00 //
0074 00F0A49C Int32 LastJump = 322853 0x4ED27 //
0078 00F0A4A0 Float LastJumpHeight = 9.500 //
007C 00F0A4A4 Int32 LastSplash = 0 0x0 //
0080 00F0A4A8 UInt8 Move = 0 0x0 //
0081 00F0A4A9 UInt8 Strafe = 0 0x0 //
0082 00F0A4AA UInt8 State = 0 0x0 //
```

ac_client.exe (ID: 38192)

Daniel: Web Attack/Defense

- Key Questions: To what extent does modern frameworks defend popular web attacks?
- Web service: MySQL/Springboot/Vue
- Division of Responsibility: DTO/DAO; Service; Controller/VO
- Auth Control + SpringSecurity
- Failed Attacks: CSRF; SQLi; Injection

Daniel: Web Attack/Defense



2017

A01:2017-Injection

A02:2017-Broken Authentication

A03:2017-Sensitive Data Exposure

A04:2017-XML External Entities (XXE)

A05:2017-Broken Access Control

A06:2017-Security Misconfiguration

A07:2017-Cross-Site Scripting (XSS)

A08:2017-Insecure Deserialization

A09:2017-Using Components with Known Vulnerabilities

A10:2017-Insufficient Logging & Monitoring

2021

A01:2021-Broken Access Control

A02:2021-Cryptographic Failures

A03:2021-Injection

(New) A04:2021-Insecure Design

A05:2021-Security Misconfiguration

A06:2021-Vulnerable and Outdated Components

A07:2021-Identification and Authentication Failures

(New) A08:2021-Software and Data Integrity Failures

A09:2021-Security Logging and Monitoring Failures*

(New) A10:2021-Server-Side Request Forgery (SSRF)*

* From the Survey

Next Meetings

Next Thursday:

- ROP
- In person in Siebel 1404 at 6pm
- Learn how to bypass non executable stack
- Background in pwn/asm from last semester very helpful

Next Seminar:

- UIUCTF planning