hackersvilla.xyz

Cracking Softwares
Diving into RE

Presenter – Sanchay Singh
@ THM Delhi, eSec Forte 19 May 2024

$ whoami_

hackersvilla.xyz

- -> **Co-founder of HackersVilla CyberSecurity**
- -> *Security Consultant/Trainer at MakeIntern*
- -> *Working as SME with UpgradCampus*
- -> *Trained Employees of KPMG, Cognizant, etc*
- -> *Security Mentor at OWASP Delhi & BSides Noida*
- -> *Speaker at BSides, Defcon Delhi, CRACCon, etc*
- -> *Active part of NULL , CRAC, THM Delhi Chapter*

Sanchay Singh

CYBERSECURITY EXPERT | CORPORATE TRAINER | PUBLIC SPEAKER

sanchayofficial

sanchayofficial@gmail.com

My Journey

Welcome to the Software Cracking Session

Agenda Overview

1. Understanding ELF/PE based Executables
2. Understanding Static and Dynamic Analysis
3. Live Demonstration on PE using win32dbg
4. Microsoft Key Bypassing (Key Extraction and SL Manager)
5. Adobe Key Bypassing (DLL Injections)
6. Cracking into VideoGames and creating hacks

Prerequisites for Participants

- Intermediate level of Cybersecurity Knowledge
- A working laptop/system (to follow along)
- Curiosity and Enthusiasm

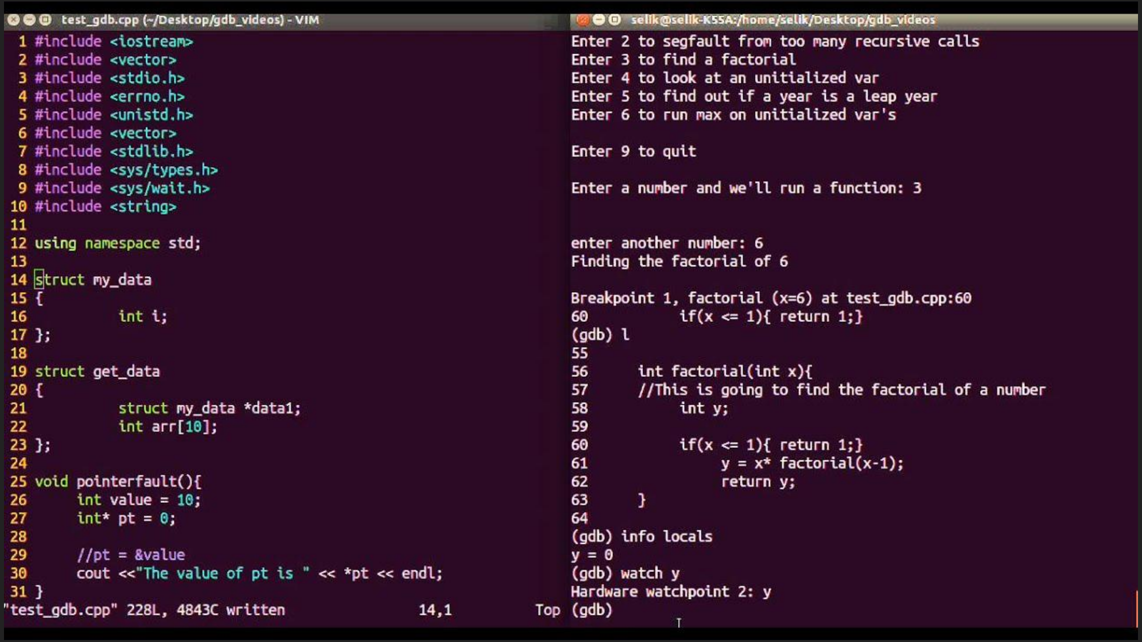
hackersvilla.xyz

EXE/ELF/PE/Mach-O Executables

Reverse Engineering and Analysis

hackersvilla.xyz

Static Analysis of a Binary

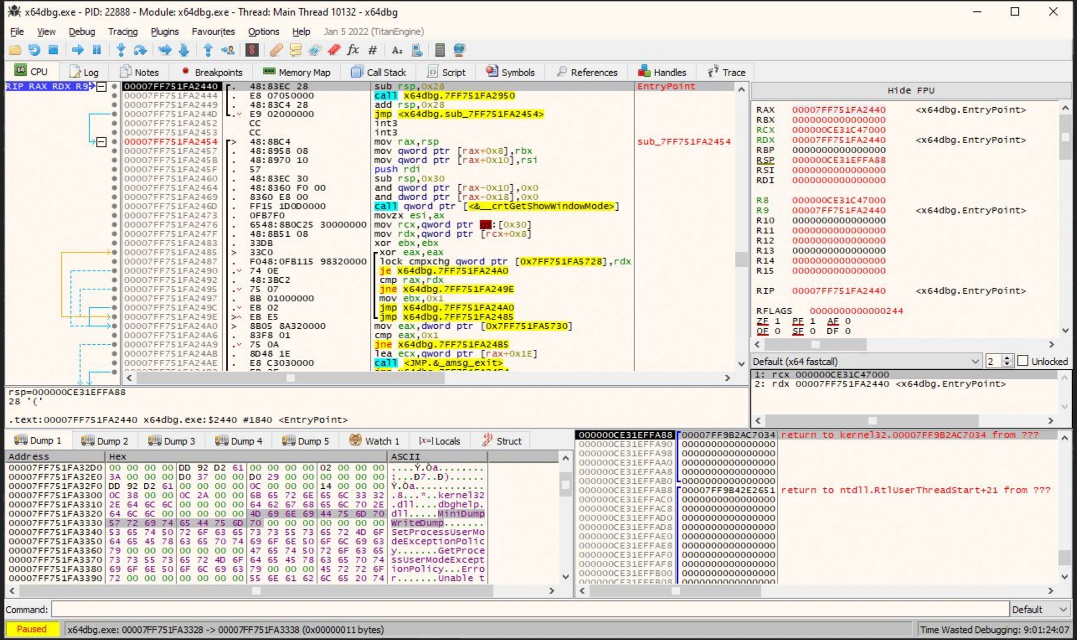

test_gdb.cpp (~/Desktop/gdb_videos) - VIM
selik@selik-K55A:/home/selik/Desktop/gdb_videos

```cpp
#include <iostream>
#include <vector>
#include <stdio.h>
#include <errno.h>
#include <unistd.h>
#include <vector>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <string>

using namespace std;

struct my_data
{
        int i;
};

struct get_data
{
        struct my_data *data1;
        int arr[10];
};

void pointerfault(){
    int value = 10;
    int* pt = 0;

    //pt = &value
    cout <<"The value of pt is " << *pt << endl;
}
"test_gdb.cpp" 228L, 4843C written                 14,1          Top
```

```
Enter 2 to segfault from too many recursive calls
Enter 3 to find a factorial
Enter 4 to look at an unitialized var
Enter 5 to find out if a year is a leap year
Enter 6 to run max on unitialized var's

Enter 9 to quit

Enter a number and we'll run a function: 3


enter another number: 6
Finding the factorial of 6

Breakpoint 1, factorial (x=6) at test_gdb.cpp:60
60              if(x <= 1){ return 1;}
(gdb) l
55
56          int factorial(int x){
57          //This is going to find the factorial of a number
58              int y;
59
60              if(x <= 1){ return 1;}
61                  y = x* factorial(x-1);
62                  return y;
63          }
64
(gdb) info locals
y = 0
(gdb) watch y
Hardware watchpoint 2: y
(gdb)
```

hackersvilla.xyz

Dynamic Analysis of a Binary



x64dbg.exe - PID: 22888 - Module: x64dbg.exe - Thread: Main Thread 10132 - x64dbg

```
RIP RAX RDX R9  00007FF751FA2440  48:83EC 28          sub rsp,0x28                                   EntryPoint
                00007FF751FA2444  E8 07050000         call x64dbg.7FF751FA2950
                00007FF751FA2449  48:83C4 28          add rsp,0x28
                00007FF751FA244D  E9 02000000         jmp <x64dbg.sub_7FF751FA2454>
                00007FF751FA2452  CC                  int3
                00007FF751FA2453  CC                  int3
                00007FF751FA2454  48:8BC4             mov rax,rsp                                    sub_7FF751FA2454
                00007FF751FA2457  48:8958 08          mov qword ptr [rax+0x8],rbx
                00007FF751FA245B  48:8970 10          mov qword ptr [rax+0x10],rsi
                00007FF751FA245F  57                  push rdi
                00007FF751FA2460  48:83EC 30          sub rsp,0x30
                00007FF751FA2464  48:8360 F0 00       and qword ptr [rax-0x10],0x0
                00007FF751FA2469  8360 E8 00          and dword ptr [rax-0x18],0x0
                00007FF751FA246D  FF15 1D0D0000       call qword ptr [<&__crtGetShowWindowMode>]
                00007FF751FA2473  0FB7F0              movzx esi,ax
                00007FF751FA2476  6548:8B0C25 30000000 mov rcx,qword ptr gs:[0x30]
                00007FF751FA247F  48:8B51 08          mov rdx,qword ptr [rcx+0x8]
                00007FF751FA2483  33DB                xor ebx,ebx
                00007FF751FA2485  33C0                xor eax,eax
                00007FF751FA2487  F048:0FB115 98320000 lock cmpxchg qword ptr [0x7FF751FA5728],rdx
                00007FF751FA2490  74 0E               je x64dbg.7FF751FA24A0
                00007FF751FA2492  48:3BC2             cmp rax,rdx
                00007FF751FA2495  75 07               jne x64dbg.7FF751FA249E
                00007FF751FA2497  BB 01000000         mov ebx,0x1
                00007FF751FA249C  EB 02               jmp x64dbg.7FF751FA24A0
                00007FF751FA249E  EB E5               jmp x64dbg.7FF751FA2485
                00007FF751FA24A0  8B05 8A320000       mov eax,dword ptr [0x7FF751FA5730]
                00007FF751FA24A6  83F8 01             cmp eax,0x1
                00007FF751FA24A9  75 0A               jne x64dbg.7FF751FA24B5
                00007FF751FA24AB  8D48 1E             lea ecx,qword ptr [rax+0x1E]
                00007FF751FA24AE  E8 C3030000         call <JMP.&_amsg_exit>
```

```
rsp=000000CE31EFFA88
28 '('

.text:00007FF751FA2440 x64dbg.exe:$2440 #1840 <EntryPoint>
```

```
Hide FPU

RAX     00007FF751FA2440     <x64dbg.EntryPoint>
RBX     0000000000000000
RCX     000000CE31C47000
RDX     00007FF751FA2440     <x64dbg.EntryPoint>
RBP     0000000000000000
RSP     000000CE31EFFA88
RSI     0000000000000000
RDI     0000000000000000

R8      000000CE31C47000
R9      00007FF751FA2440     <x64dbg.EntryPoint>
R10     0000000000000000
R11     0000000000000000
R12     0000000000000000
R13     0000000000000000
R14     0000000000000000
R15     0000000000000000

RIP     00007FF751FA2440     <x64dbg.EntryPoint>

RFLAGS    0000000000000244
ZF 1   PF 1   AF 0
OF 0   SF 0   DF 0
```

Default (x64 fastcall) 2 Unlocked

```
1: rcx 000000CE31C47000
2: rdx 00007FF751FA2440 <x64dbg.EntryPoint>
```

```
Address           Hex                                               ASCII
00007FF751FA32D0  00 00 00 00 DD 92 D2 61 00 00 00 00 02 00 00 00  ....Ý.Òa........
00007FF751FA32E0  3A 00 00 00 D0 37 00 00 D0 29 00 00 00 00 00 00  :...Ð7..Ð)......
00007FF751FA32F0  DD 92 D2 61 00 00 00 00 0C 00 00 00 14 00 00 00  Ý.Òa............
00007FF751FA3300  0C 38 00 00 0C 2A 00 00 6B 65 72 6E 65 6C 33 32  .8...*..kernel32
00007FF751FA3310  2E 64 6C 6C 00 00 00 00 64 62 67 68 65 6C 70 2E  .dll....dbghelp.
00007FF751FA3320  64 6C 6C 00 00 00 00 00 4D 69 6E 69 44 75 6D 70  dll.....MiniDump
00007FF751FA3330  57 72 69 74 65 44 75 6D 70 00 00 00 00 00 00 00  WriteDump.......
00007FF751FA3340  53 65 74 50 72 6F 63 65 73 73 55 73 65 72 4D 6F  SetProcessUserMo
00007FF751FA3350  64 65 45 78 63 65 70 74 69 6F 6E 50 6F 6C 69 63  deExceptionPolic
00007FF751FA3360  79 00 00 00 00 00 00 00 47 65 74 50 72 6F 63 65  y.......GetProce
00007FF751FA3370  73 73 55 73 65 72 4D 6F 64 65 45 78 63 65 70 74  ssUserModeExcept
00007FF751FA3380  69 6F 6E 50 6F 6C 69 63 79 00 00 00 45 72 72 6F  ionPolicy...Erro
00007FF751FA3390  72 00 00 00 00 00 00 00 55 6E 61 62 6C 65 20 74  r.......Unable t
```

```
000000CE31EFFA88  00007FF9B2AC7034  return to kernel32.00007FF9B2AC7034 from ???
000000CE31EFFA90  0000000000000000
000000CE31EFFA98  0000000000000000
000000CE31EFFAA0  0000000000000000
000000CE31EFFAA8  0000000000000000
000000CE31EFFAB0  0000000000000000
000000CE31EFFAB8  00007FF9B42E2651  return to ntdll.RtlUserThreadStart+21 from ???
000000CE31EFFAC0  0000000000000000
000000CE31EFFAC8  0000000000000000
000000CE31EFFAD0  0000000000000000
000000CE31EFFAD8  0000000000000000
000000CE31EFFAE0  0000000000000000
000000CE31EFFAE8  0000000000000000
000000CE31EFFAF0  0000000000000000
000000CE31EFFAF8  0000000000000000
000000CE31EFFB00  0000000000000000
```

Command: Default

Paused x64dbg.exe: 00007FF751FA3328 -> 00007FF751FA3338 (0x00000011 bytes) Time Wasted Debugging: 9:01:24:07

hackersvilla.xyz

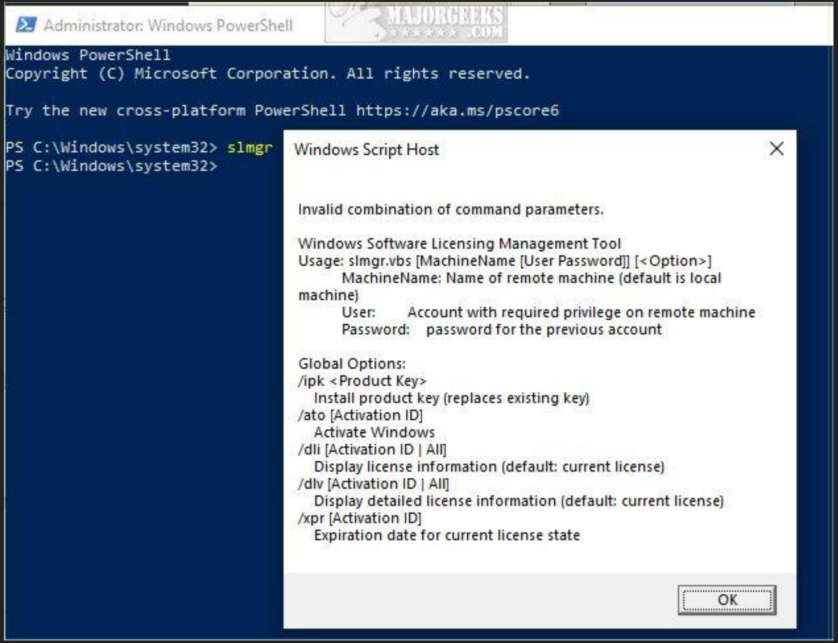
Lets learn Bypassing

DEMO TIME

Microsoft
Key Bypassing

hackersvilla.xyz

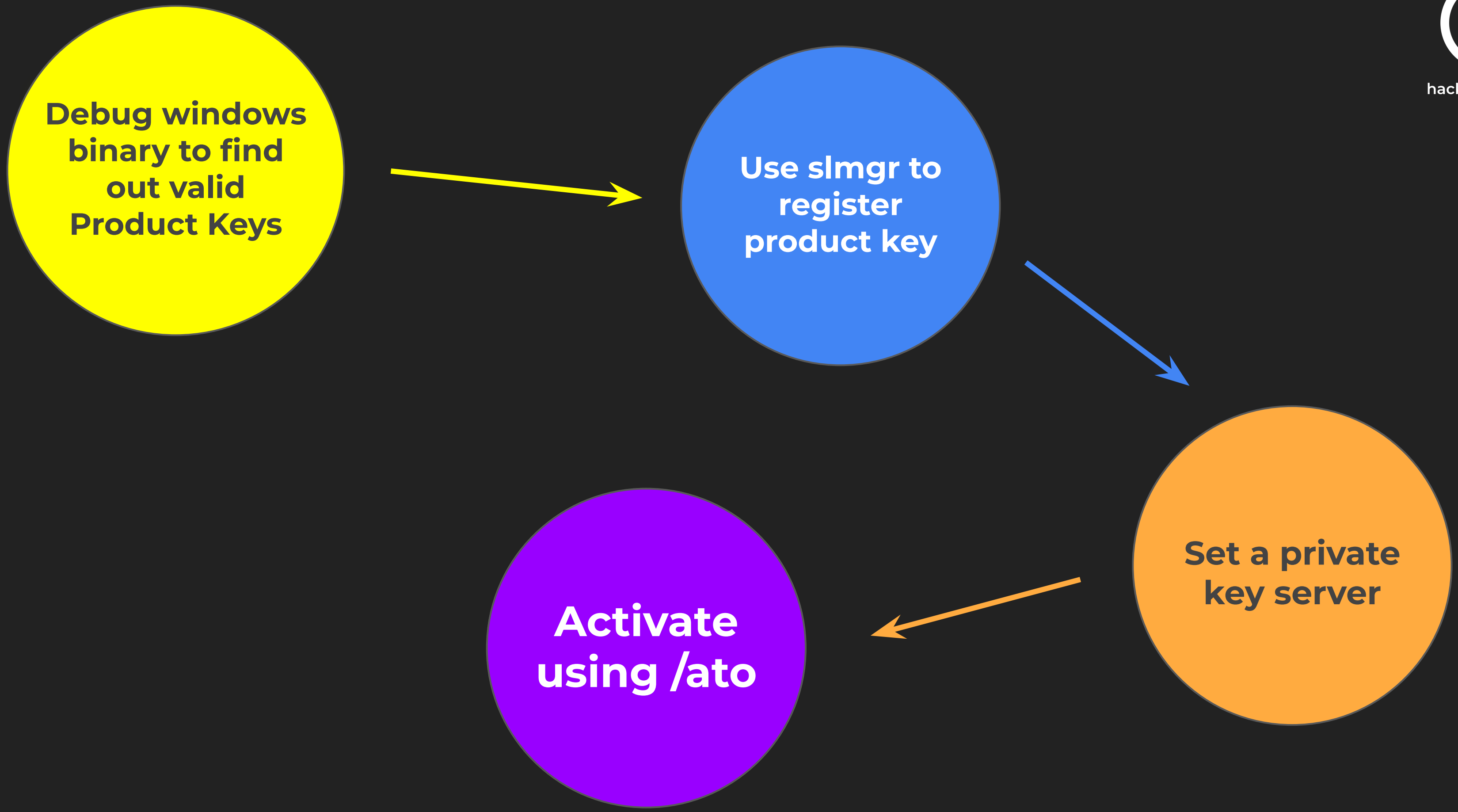
slmgr (Software Licensing Manager)



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> slmgr
PS C:\Windows\system32>
```

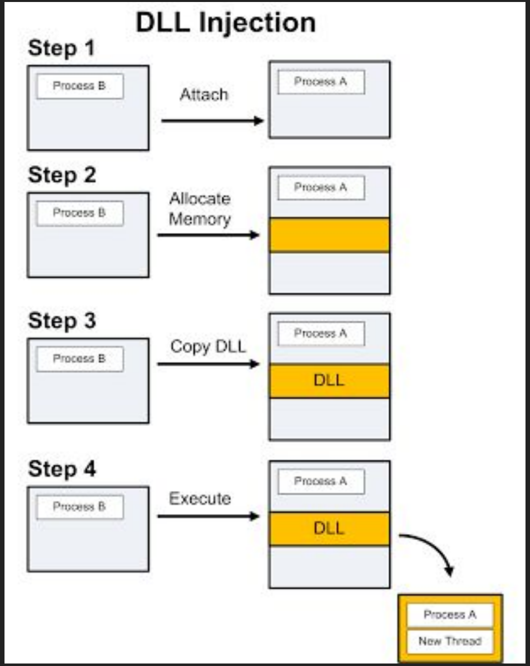
```
Invalid combination of command parameters.

Windows Software Licensing Management Tool
Usage: slmgr.vbs [MachineName [User Password]] [<Option>]
        MachineName: Name of remote machine (default is local
machine)
        User:       Account with required privilege on remote machine
        Password:    password for the previous account

Global Options:
/ipk <Product Key>
   Install product key (replaces existing key)
/ato [Activation ID]
   Activate Windows
/dli [Activation ID | All]
   Display license information (default: current license)
/dlv [Activation ID | All]
   Display detailed license information (default: current license)
/xpr [Activation ID]
   Expiration date for current license state
```

hackersvilla.xyz


Debug windows binary to find out valid Product Keys
Use slmgr to register product key
Set a private key server
Activate using /ato

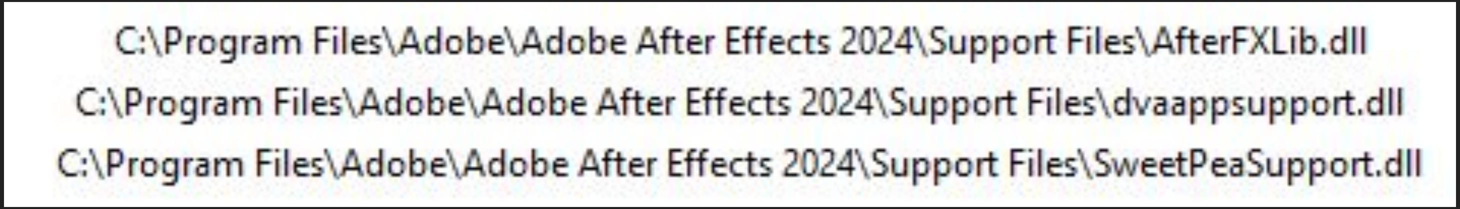
Adobe
Key Bypassing

DLL Injections



DLL injection is a classic method of putting code into another process in memory.

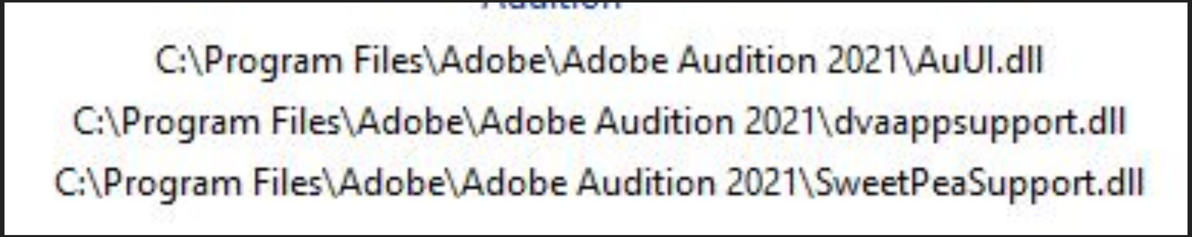
After Effects



```
C:\Program Files\Adobe\Adobe After Effects 2024\Support Files\AfterFXLib.dll
C:\Program Files\Adobe\Adobe After Effects 2024\Support Files\dvaappsupport.dll
C:\Program Files\Adobe\Adobe After Effects 2024\Support Files\SweetPeaSupport.dll
```

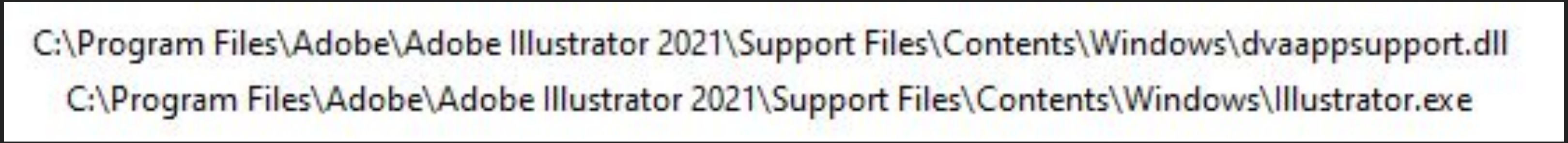
hackersvilla.xyz

Audition



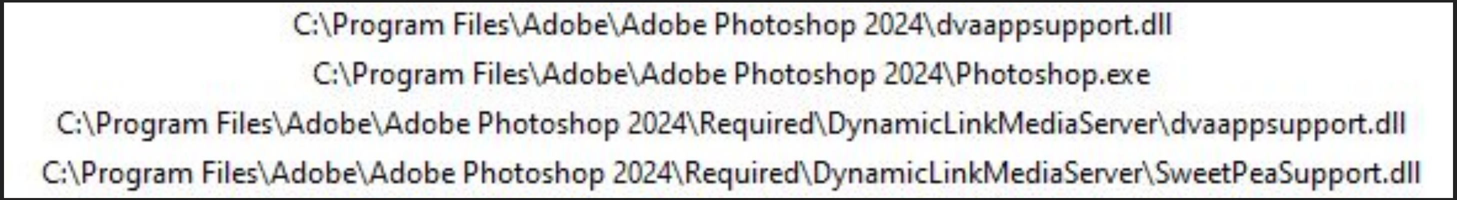
```
C:\Program Files\Adobe\Adobe Audition 2021\AuUI.dll
C:\Program Files\Adobe\Adobe Audition 2021\dvaappsupport.dll
C:\Program Files\Adobe\Adobe Audition 2021\SweetPeaSupport.dll
```

Illustrator



```
C:\Program Files\Adobe\Adobe Illustrator 2021\Support Files\Contents\Windows\dvaappsupport.dll
C:\Program Files\Adobe\Adobe Illustrator 2021\Support Files\Contents\Windows\Illustrator.exe
```

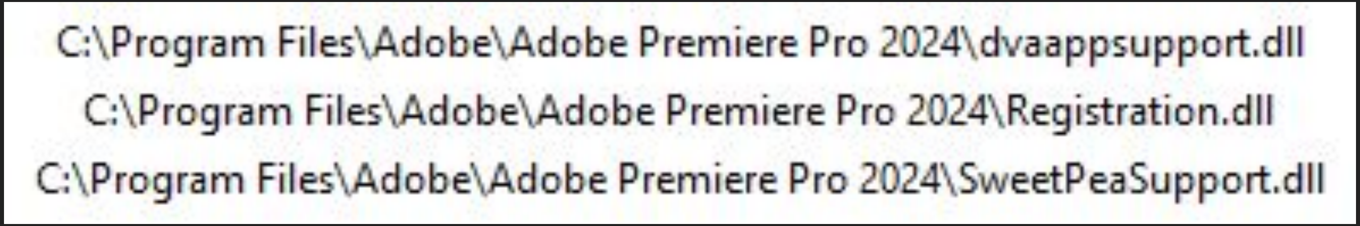
Photoshop



```
C:\Program Files\Adobe\Adobe Photoshop 2024\dvaappsupport.dll
C:\Program Files\Adobe\Adobe Photoshop 2024\Photoshop.exe
C:\Program Files\Adobe\Adobe Photoshop 2024\Required\DynamicLinkMediaServer\dvaappsupport.dll
C:\Program Files\Adobe\Adobe Photoshop 2024\Required\DynamicLinkMediaServer\SweetPeaSupport.dll
```

hackersvilla.xyz

Premiere Pro



```
C:\Program Files\Adobe\Adobe Premiere Pro 2024\dvaappsupport.dll
C:\Program Files\Adobe\Adobe Premiere Pro 2024\Registration.dll
C:\Program Files\Adobe\Adobe Premiere Pro 2024\SweetPeaSupport.dll
```

Game
Cracking & Hacking

Potential Vulnerabilities

Unreal Engine

- Code Injection
- Remote Code Execution (RCE)
- Exposed APIs
- Insecure File Handling

Unity

- Insecure Asset Store Content
- Data Exposure in WebGL Builds
- Cross-Site Scripting (XSS)
- Denial of Service (DoS) Attacks

hackersvilla.xyz

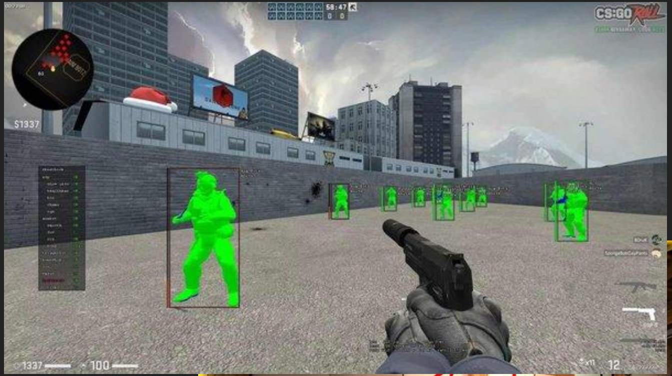
Common Security Challenges in Video Games

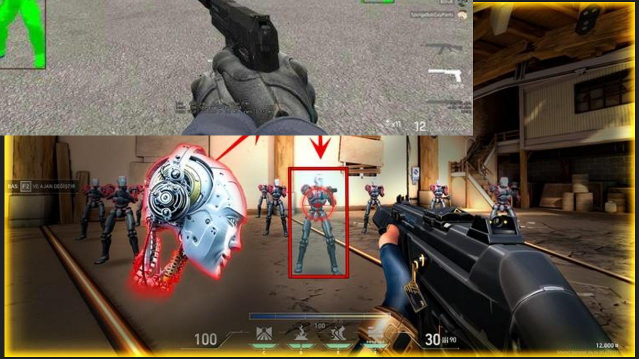
hackersvilla.xyz

Analysis of Common Threats

Aimbots and Wallhacks:

- Players using aimbots and wallhacks disrupt fair play.
- Real-world example: A popular first-person shooter faced widespread cheating issues, impacting the gaming experience for honest players.



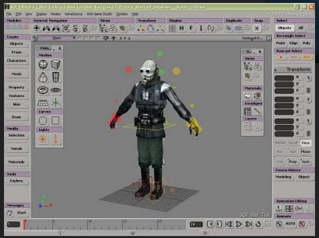


hackersvilla.xyz

Analysis of Common Threats

Risks of User-Generated Content

- User-generated content, while enriching the gaming experience, poses risks
- Example: A modding community unintentionally introduced a mod that compromised player privacy by accessing unintended game data

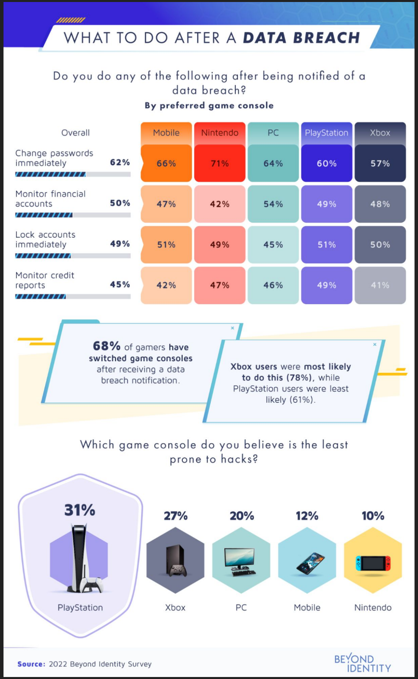


hackersvilla.xyz

Analysis of Common Threats

Account Breaches and Privacy Concerns

- Unauthorized access to player accounts can lead to data breaches and privacy concerns.
- Example: A major gaming platform experienced a security incident resulting in unauthorized access to millions of user accounts.



WHAT TO DO AFTER A ***DATA BREACH***

Do you do any of the following after being notified of a data breach?

By preferred game console

	Overall	Mobile	Nintendo	PC	PlayStation	Xbox
Change passwords immediately	62%	66%	71%	64%	60%	57%
Monitor financial accounts	50%	47%	42%	54%	49%	48%
Lock accounts immediately	49%	51%	49%	45%	51%	50%
Monitor credit reports	45%	42%	47%	46%	49%	41%

68% of gamers **have switched game consoles** after receiving a data breach notification.

Xbox users were **most likely to do this (78%)**, while PlayStation users were least likely (61%).

Which game console do you believe is the least prone to hacks?

PlayStation	Xbox	PC	Mobile	Nintendo
31%	27%	20%	12%	10%

Source: 2022 Beyond Identity Survey

BEYOND IDENTITY

hackersvilla.xyz

Some Resources to help you learn

Reverse Engineering Challenges:

challenges.re

Game Hacking Tuts

hackersvilla.xyz

So….
Questions?

hackersvilla.xyz

Thank You