Concordia Institute for Information System Engineering (CIISE)
Concordia University

INSE 6140 Malware Defenses and Application Security

Project Area 3 : DLL Injection Detection using Ghidra.

Submitted to: **Professor Dr. Makan Pourzandi**

Submitted by:

Student Name	**Student ID**
Aniket Agarwal	40266485
Kalyani Batle	40243967
Aathira Dineshan	40270695

Introduction

Problem Statement:

- DLL injection common among game hackers
- Need to detect the DLL injection

Motivation:

- Ghidra
  - Open source
  - Integrate of scripts and plugins
- Script to detect DLL injection

Objective:

Detect potential DLL injection attacks and warn the users

Steps: DLL Injection Attack

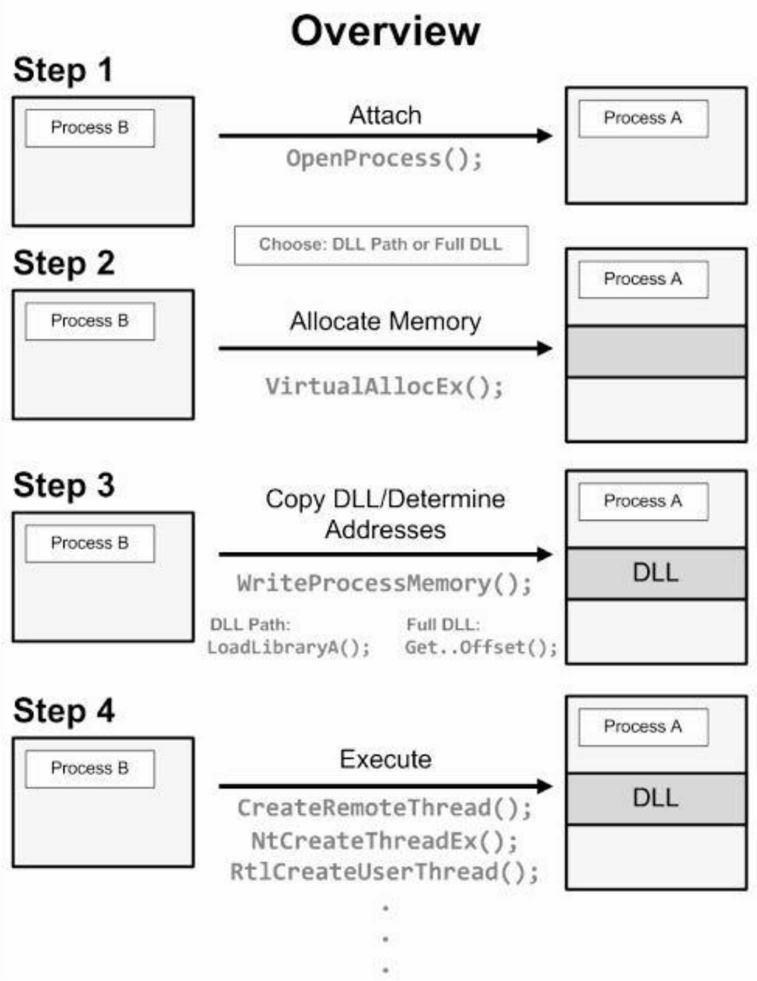
Game chosen: Wesnoth

1. Creating the DLL Project
2. Implementing DllMain Function
3. Injecting DLL into Wesnoth game using DLL injector executable
4. Creating parallel threads in running game
5. Testing

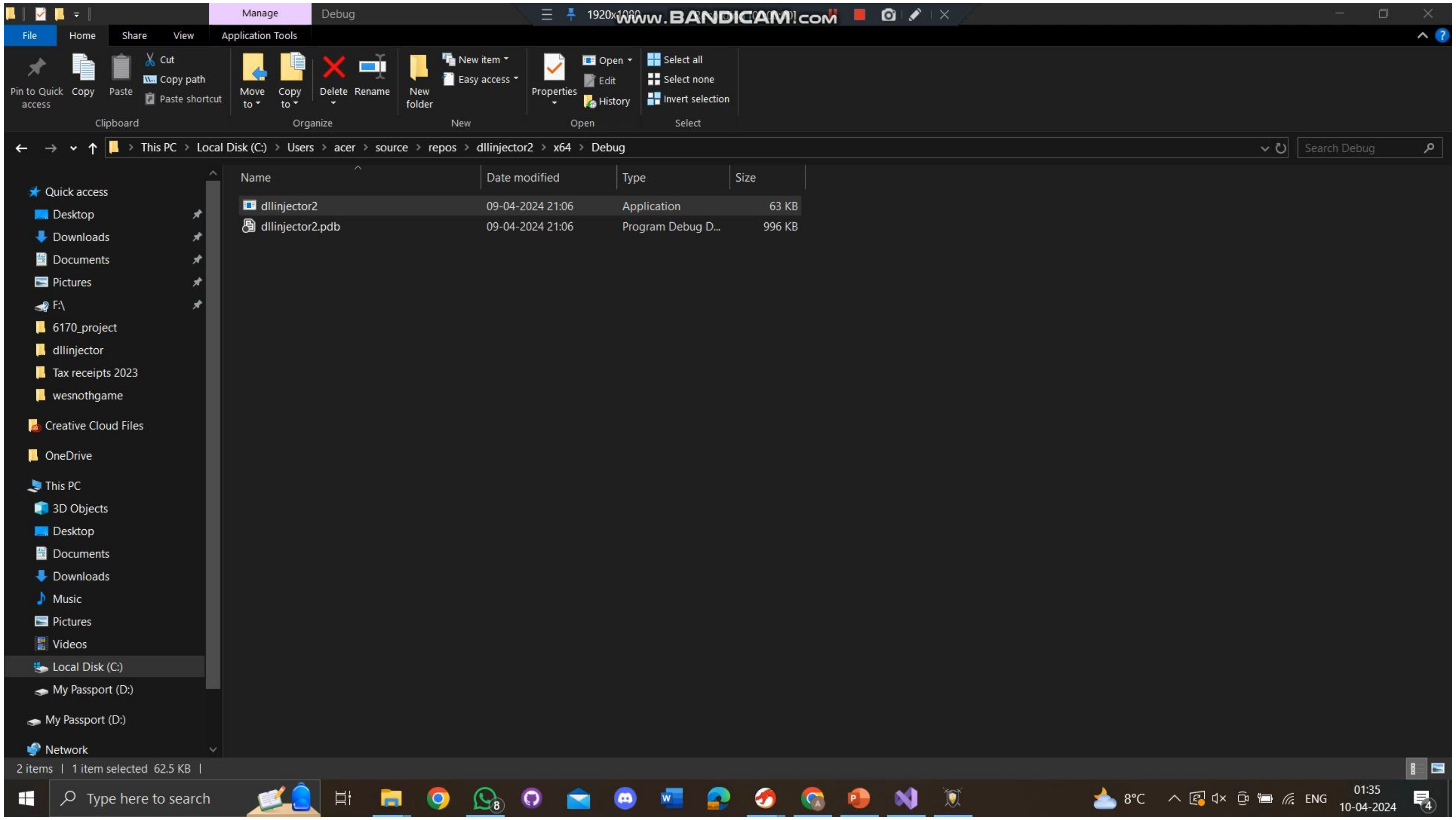
DLL Injector Basic Structure

Common functions:

1. **OpenProcess**
2. **VirtualAllocEX**
3. **WriteProcessMemory**
4. GetModuleHandle
5. **CreateRemoteThread**
6. VirtualFreeEx
7. **CloseHandle**



DLL Injection Attack Demo



This PC > Local Disk (C:) > Users > acer > source > repos > dllinjector2 > x64 > Debug

Name	Date modified	Type	Size
dllinjector2	09-04-2024 21:06	Application	63 KB
dllinjector2.pdb	09-04-2024 21:06	Program Debug D...	996 KB

2 items | 1 item selected 62.5 KB |

Algorithm of detection script

Algorithm Description:

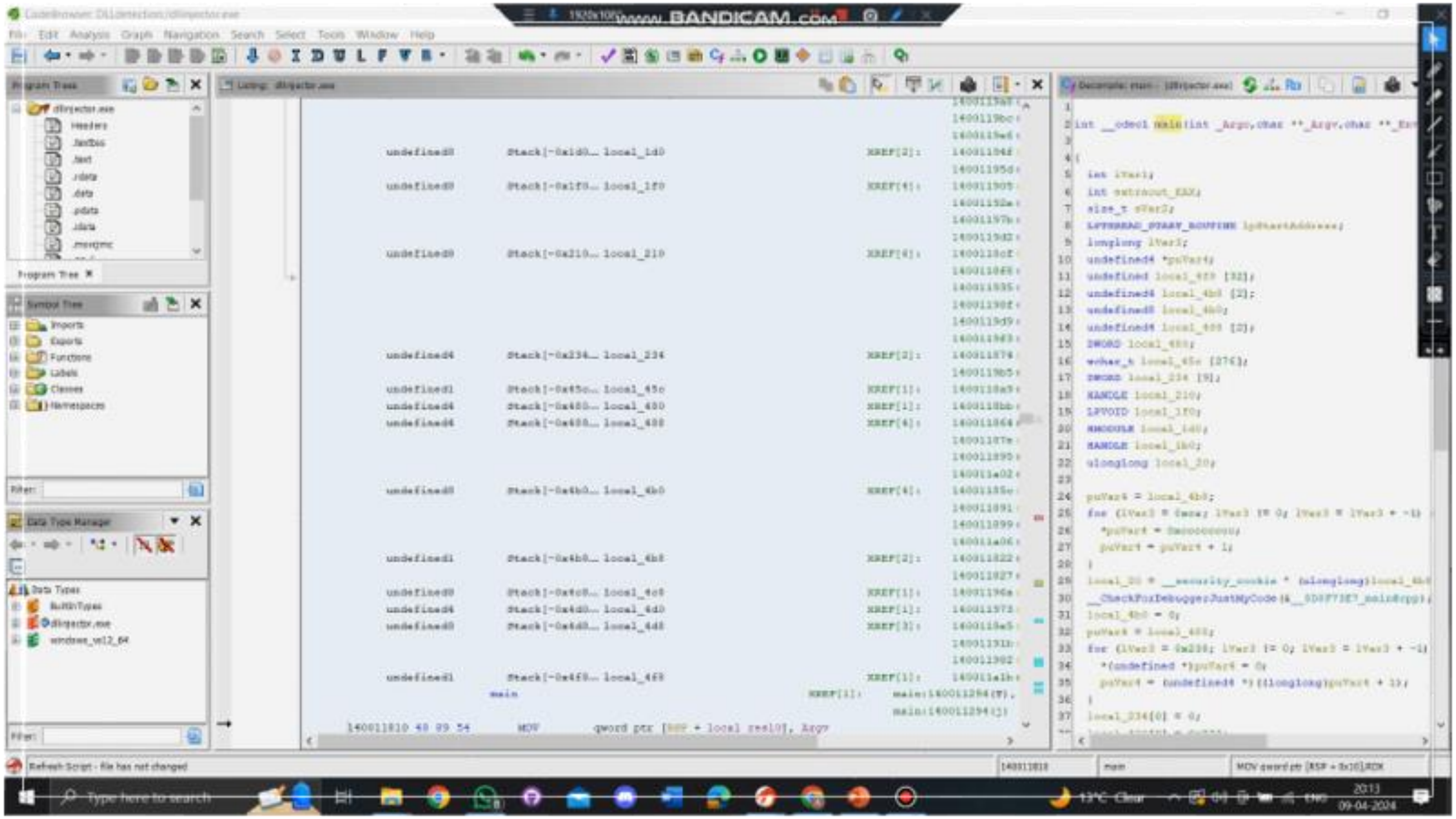
- Scans current program to analyze assembly instructions in a binary executable within Ghidra
- Get current function address and instruction address
- Iterate over instruction and check for "CALL" instruction
- Check for target function by extracting called functions symbol
- If all match
    - Print "This binary maybe vulnerable to DLL Injection attack" on console
    - Highlight the function calls in Listing window

Jython script executed within Ghidra script manager.

Target Functions

- OpenProcess
- WriteProcessMemory
- CreateRemoteThread
- VirtualAlloc
- CloseHandle

Proof of concept: Script Demo



```
             undefined8     Stack[-0x1d0... local_1d0                    XREF[2]:   14001194f
                                                                                    14001195d
             undefined8     Stack[-0x1f0... local_1f0                    XREF[4]:   140011905
                                                                                    14001192e
                                                                                    14001197b
                                                                                    1400119d2
             undefined8     Stack[-0x210... local_210                    XREF[4]:   1400118cf
                                                                                    1400118e8
                                                                                    140011935
                                                                                    14001198f
                                                                                    1400119d9
                                                                                    1400119e3
             undefined4     Stack[-0x234... local_234                    XREF[2]:   140011874
                                                                                    1400119b5
             undefined1     Stack[-0x45c... local_45c                    XREF[1]:   1400118a9
             undefined4     Stack[-0x480... local_480                    XREF[1]:   1400118bb
             undefined4     Stack[-0x488... local_488                    XREF[4]:   140011864
                                                                                    14001187e
                                                                                    140011895
                                                                                    140011a02
             undefined8     Stack[-0x4b0... local_4b0                    XREF[4]:   14001185c
                                                                                    140011891
                                                                                    140011899
                                                                                    140011a06
             undefined1     Stack[-0x4b8... local_4b8                    XREF[2]:   140011822
                                                                                    140011827
             undefined8     Stack[-0x4c8... local_4c8                    XREF[1]:   14001196a
             undefined4     Stack[-0x4d0... local_4d0                    XREF[1]:   140011973
             undefined8     Stack[-0x4d8... local_4d8                    XREF[3]:   1400118e5
                                                                                    14001191b
                                                                                    140011902
             undefined1     Stack[-0x4f8... local_4f8                    XREF[1]:   140011a1b
                      main                                      XREF[1]:   main:140011294(T),
                                                                           main:140011294(j)
        140011810 48 89 54         MOV         qword ptr [RSP + local_res10], Argv
```

```
int __cdecl main(int _Argc,char **_Argv,char **_Env

{
  int iVar1;
  int extraout_EAX;
  size_t sVar2;
  LPTHREAD_START_ROUTINE lpStartAddress;
  longlong lVar3;
  undefined4 *puVar4;
  undefined local_4f8 [32];
  undefined4 local_4b8 [2];
  undefined8 local_4b0;
  undefined4 local_488 [2];
  DWORD local_480;
  wchar_t local_45c [276];
  DWORD local_234 [9];
  HANDLE local_210;
  LPVOID local_1f0;
  HMODULE local_1d0;
  HANDLE local_1b0;
  ulonglong local_20;

  puVar4 = local_4b8;
  for (lVar3 = 0xe; lVar3 != 0; lVar3 = lVar3 + -1) {
    *puVar4 = 0xcccccccc;
    puVar4 = puVar4 + 1;
  }
  local_20 = __security_cookie ^ (ulonglong)local_4b8
  __CheckForDebuggerJustMyCode(&__5D8F73E7_main@cpp);
  local_4b0 = 0;
  puVar4 = local_488;
  for (lVar3 = 0x238; lVar3 != 0; lVar3 = lVar3 + -1) {
    *(undefined *)puVar4 = 0;
    puVar4 = (undefined4 *)((longlong)puVar4 + 1);
  }
  local_234[0] = 0;
```

Refresh Script - file has not changed | 140011810 | main | MOV qword ptr [RSP + 0x10],RDX

Numerical Results Based on Tested DLL Injector Executables

- **Detection Rate**: 80%
- **False Negative Rate**: 2/10 DLL injector executable not detected, 20%.

Conclusion

- Innovative Detection Script for Ghidra DLL analysis.
- High Accuracy and Low False Negative

Future Scope

- Advanced Detection Techniques
- Integration with Threat Intelligence : Explore new algorithms, ML models, Deep learning and behavior analysis methods.
- High Accuracy and Low False Positives
- Integration with Security Tools
- Incident Response and Mitigation
- User Education and Awareness

THANK YOU