

LAB16: Registry Forensics with RegRipperPlug-ins

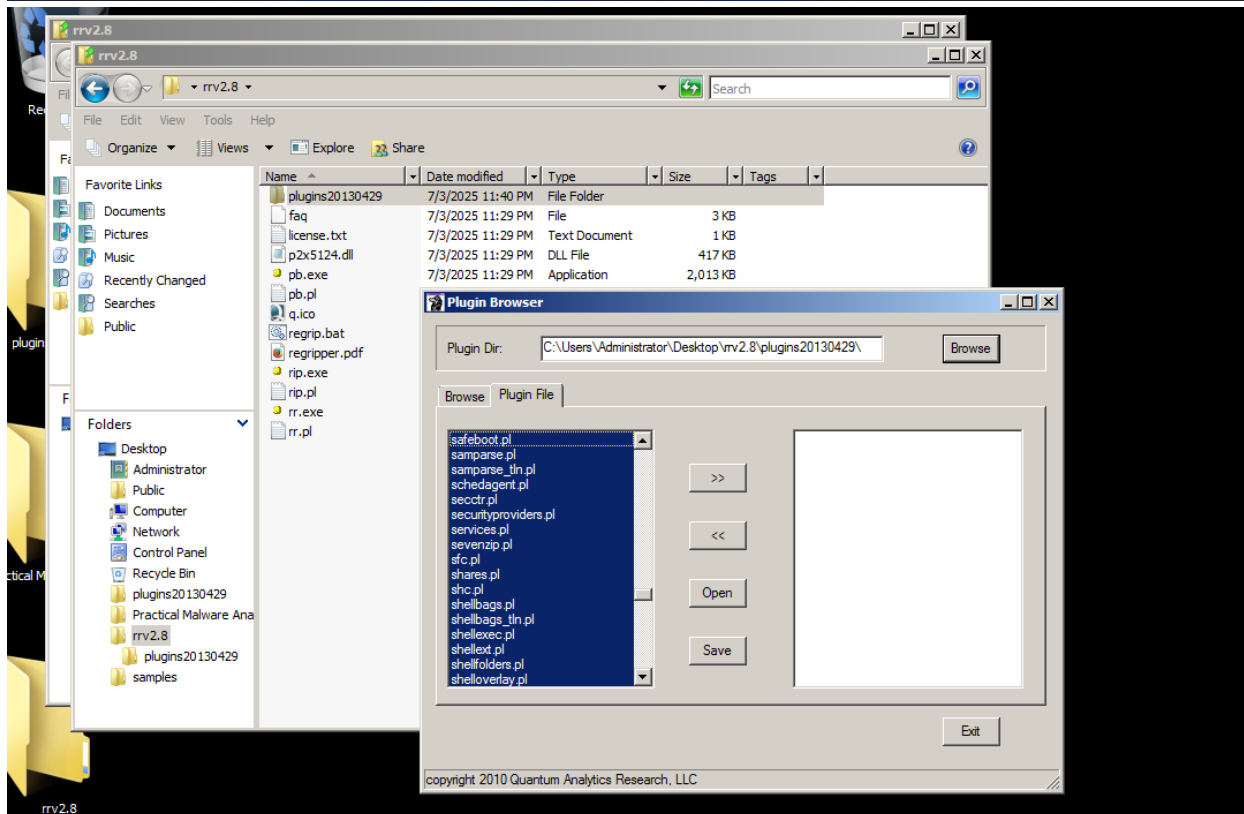
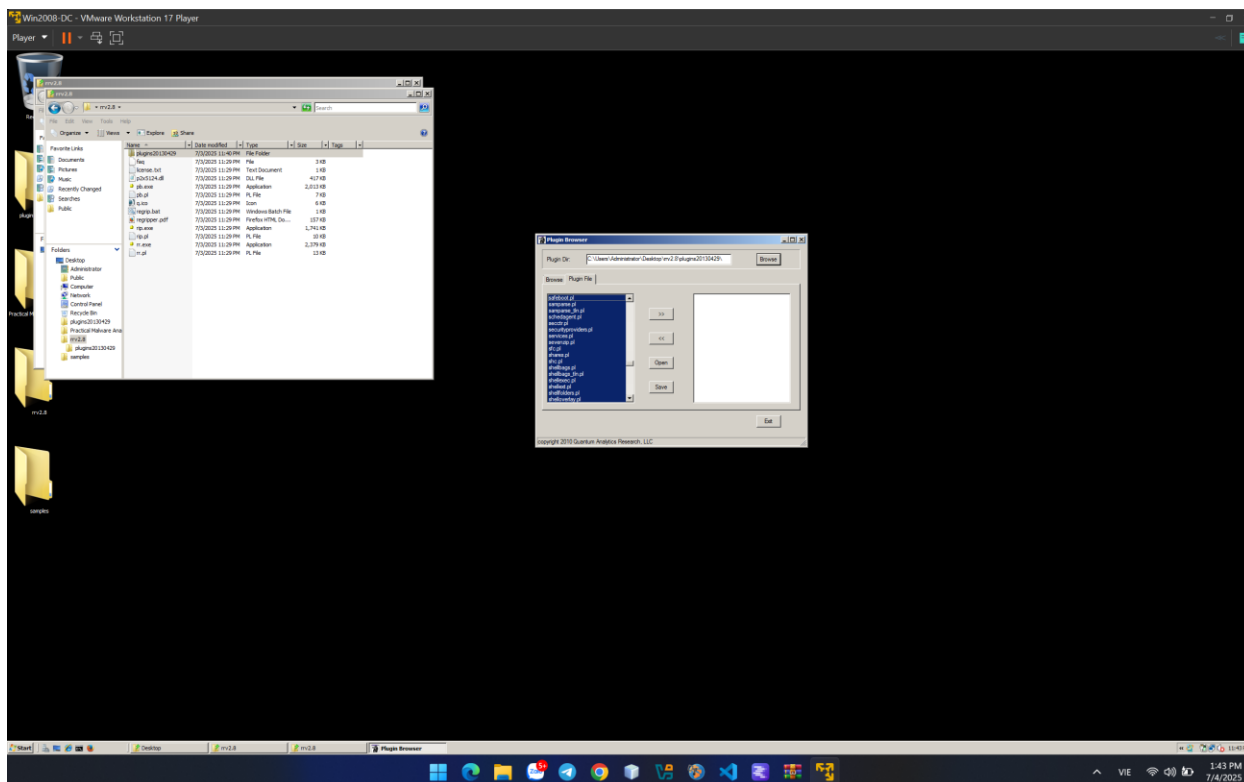
RegRipper:

You need to unzip all 3 files above Note*:

if you want to start RegRipper software, you must first import the plugins file. There will be 2 ways to import:

- Method 1: You copy and paste directly into the file rrv2.8
- Method 2: In the file rrv2.8 there is a program named "pb" and there it will display the interface window so that you can import the plugin's files.

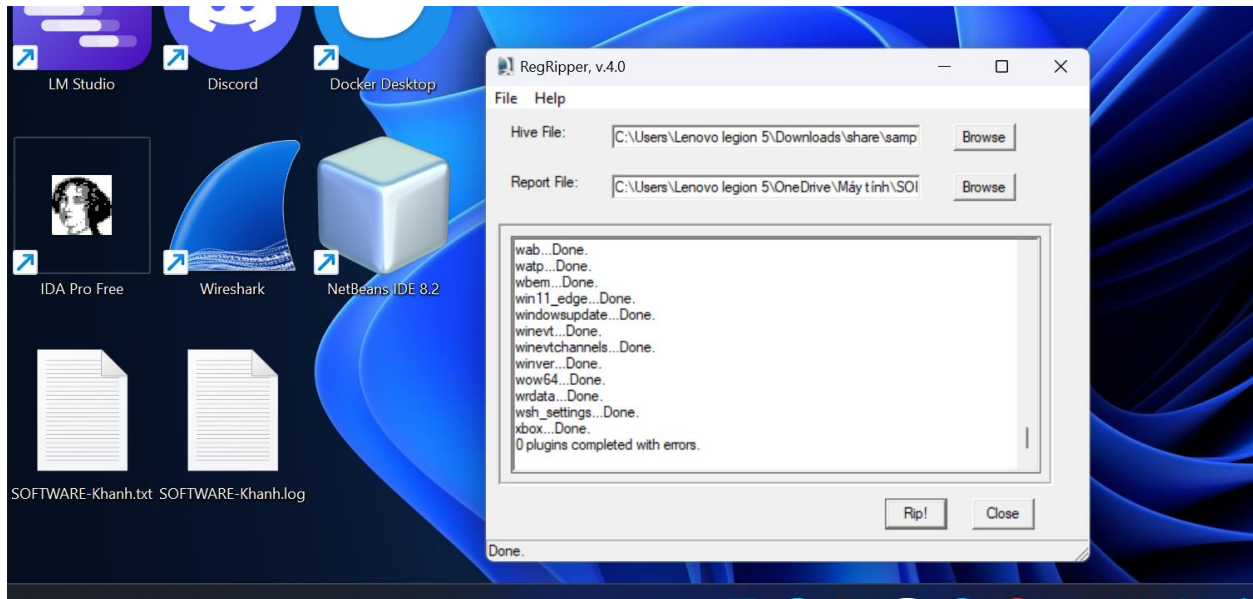
NGUYEN NAM KHANH – HE191159 – IA1902 – IAM302



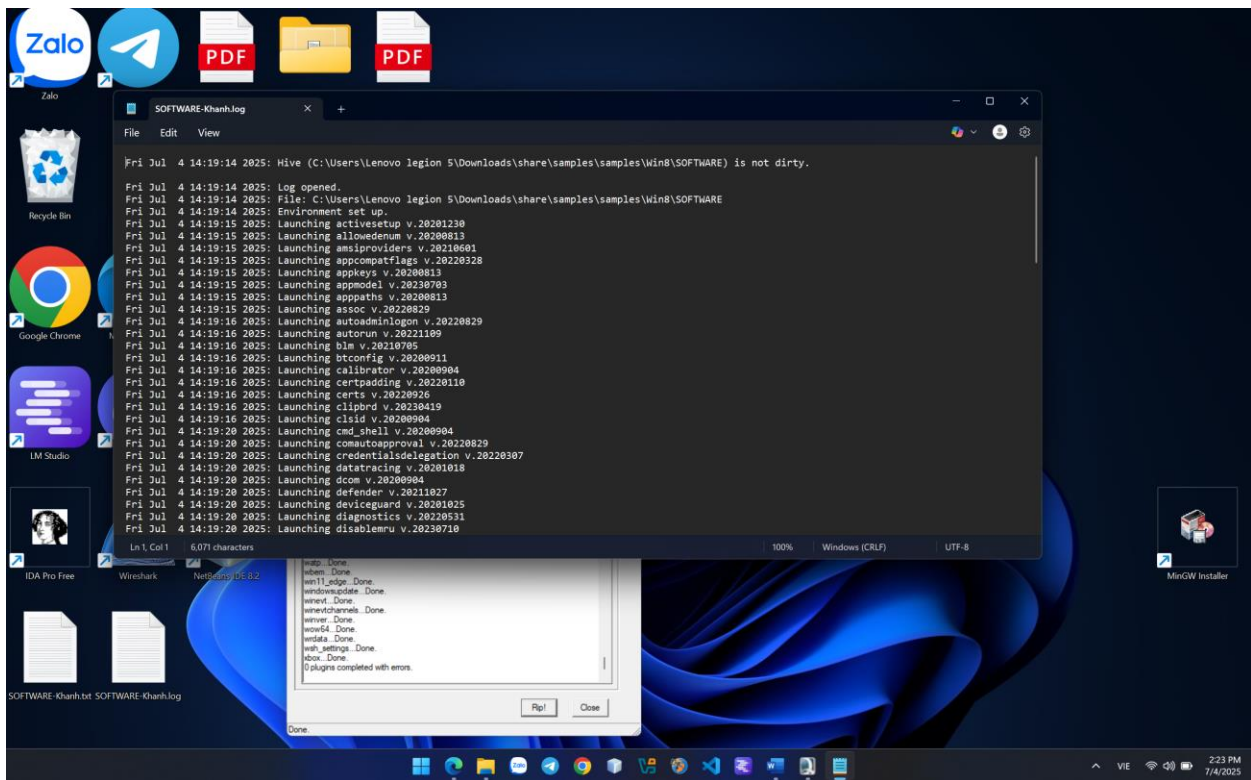
Unfortunately, after using vmware win 2008 and import plugin file to regripper 2.8 unsuccessfully, I change to my real computer and update version to 4.0 and it import automatically plugin file.

And here we will conduct the analysis:

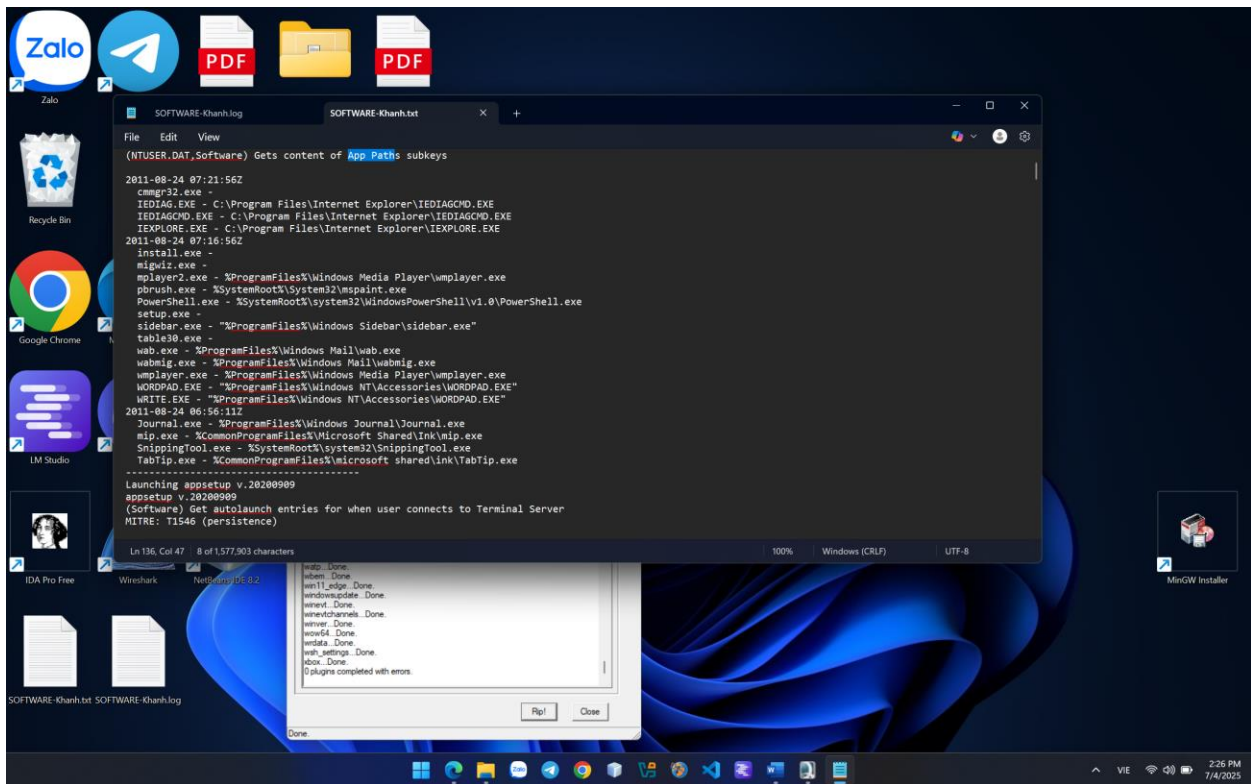
Analyst file Software:



File log:

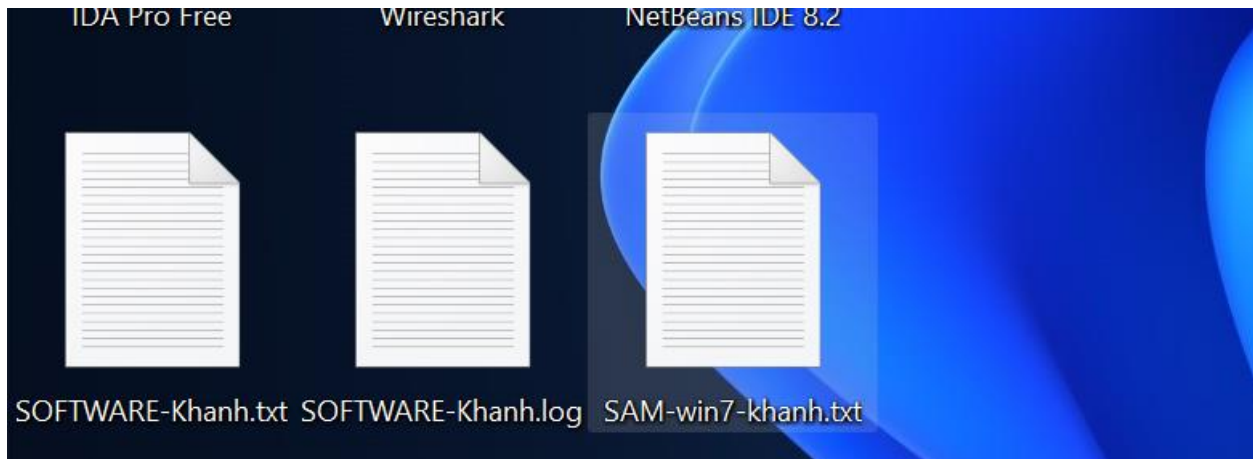


In the second file, we can easily recognize the .exe files and its location in Win 8



Analyst file SAM:

And next we will continue to analyze the SAM file in Windows 7. As you all know, the SAM file is the file that stores the user's and group membership's password, information, etc ...



NGUYEN NAM KHANH – HE191159 – IA1902 – IAM302

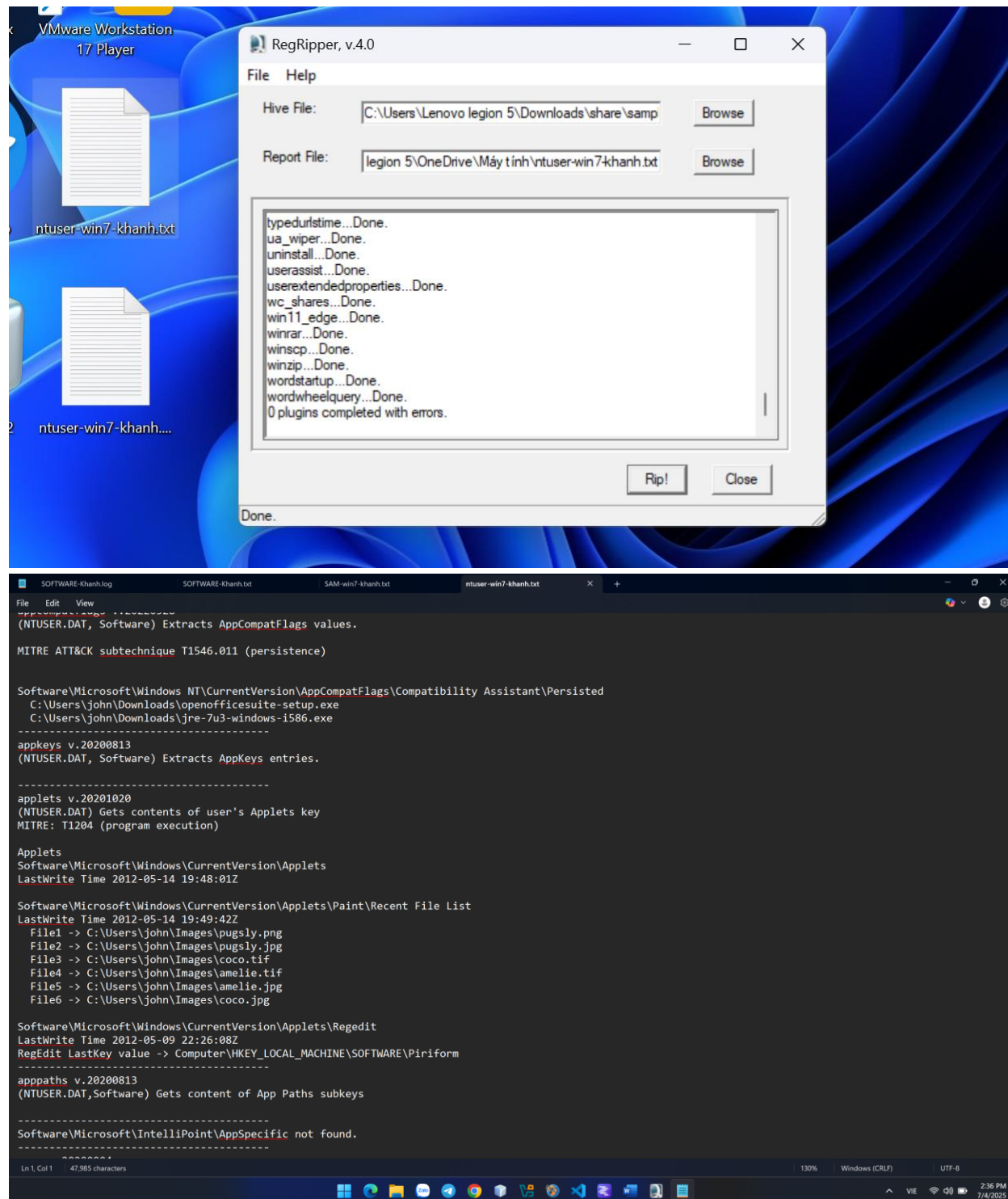
The screenshot displays a Windows desktop environment. In the background, a Notepad++ window titled 'SOFTWARE-khanh.log' is open, showing the output of a SAM file parser. The output is divided into sections for 'User Information' and 'Group Membership Information'. The 'User Information' section lists details for three users: Administrator [500], Guest [501], and John [1001]. The 'Group Membership Information' section lists details for several groups, including Users [4], Event Log Readers [0], Guests [1], Distributed COM Users [0], Administrators [3], Network Configuration Operators [0], Cryptographic Operators [0], and Power Users [0].

In the foreground, a RingRipper v4.0 application window is open. It has a 'File' menu and a 'Help' menu. The 'Hive File' field is set to 'C:\Users\Lenovo\legion 5\Downloads\share\lamp' and the 'Report File' field is set to 'C:\Users\Lenovo\legion 5\OneDrive\Máy tính\SAM-win7-khanh.log'. The application is running a process, and a status window shows the following progress: 'Hive is not dirty.', 'Logging to C:\Users\Lenovo\legion 5\OneDrive\Máy tính\SAM-win7-khanh.log', 'Hive type: sam', 'Getting list of plugins based on hive type...', 'Done', 'Start ripping...', 'sample... Done', and '0 plugins completed with errors.' The application window has a 'Rip!' button and a 'Close' button.

The taskbar at the bottom of the screen shows various application icons, including the Start button, File Explorer, and several web browsers. The system tray in the bottom right corner shows the time as 2:29 PM on 7/4/2023.

Analyst file NTUSER.DAT:

And finally the file that I want to demo for you is the ntuser.dat file. This is a setup file that configures the user's kernels or may include user accounts.



CRACKME 17:

This crackme is written in Visual Basic 6 and compiled as a p-code executable.

The objectives of this crackme are:

1. patch the file so that no matter what name or serial key you enter, it will become registered
2. create a keygen for it

1.

Find string reference of REGISTERED status

| | | | |
|----------|---|----------|------------------------------------|
| 004027E5 | mov edx,crackme17-native.401F38 | 00401F38 | L"49582F35A94345" |
| 00402908 | mov dword ptr ss:[ebp-A0],crackme17-native.401FBC | 00401FBC | L"Error" |
| 00402923 | mov dword ptr ss:[ebp-90],crackme17-native.401F6C | 00401F6C | L"Name should be more than 4 chara |
| 00402A25 | push crackme17-native.401FCC | 00401FCC | L"REGISTERED" |
| 00402A78 | mov dword ptr ss:[ebp-A0],crackme17-native.40202C | 0040202C | L"Congrats" |
| 00402A93 | mov dword ptr ss:[ebp-90],crackme17-native.401FF8 | 00401FF8 | L"Successful registration" |
| 00402AE8 | push crackme17-native.402044 | 00402044 | L"UN-REGISTERED" |

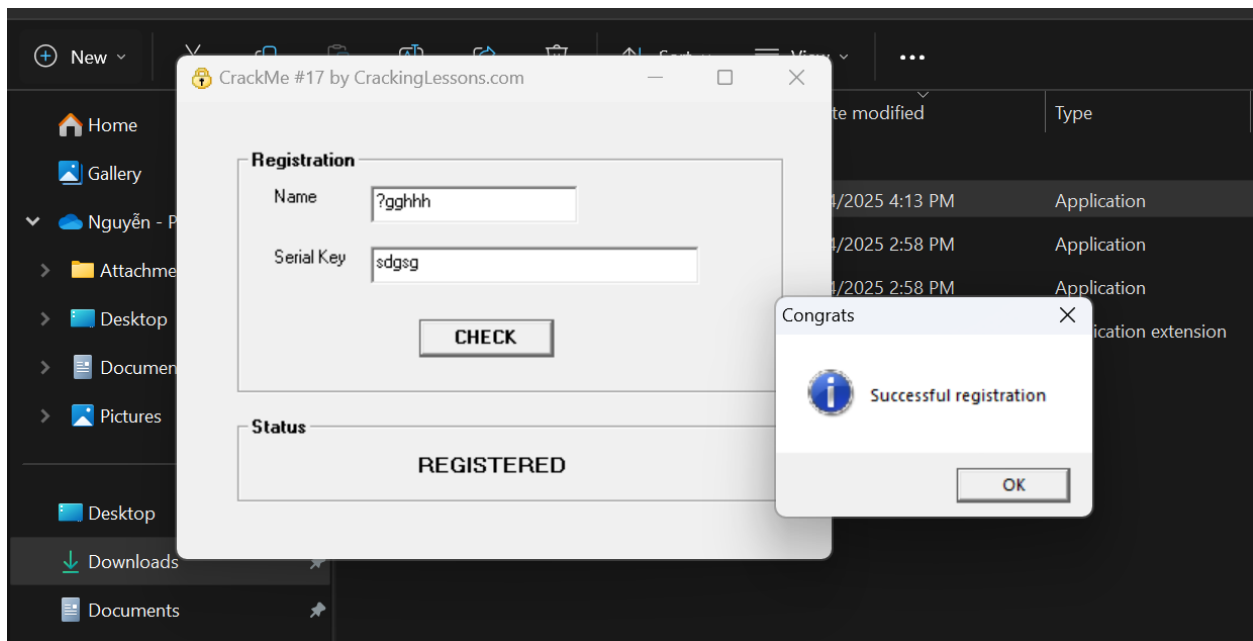
Found that there is a JE jump through REGISTERED status to UN-REGISTERED

| | | | |
|--|---------------|----------------------------------|----------------------|
| | 83C4 18 | add esp,18 | |
| | 66:3BFB | cmp di,bx | |
| | 0F84 C3000000 | je crackme17-native.402AD2 | |
| | 8B16 | mov edx,dword ptr ds:[esi] | |
| | 56 | push esi | |
| | FF92 14030000 | call dword ptr ds:[edx+314] | |
| | 50 | push eax | |
| | 8D45 D8 | lea eax,dword ptr ss:[ebp-28] | |
| | 50 | push eax | |
| | FF15 20104000 | call dword ptr ds:[<_vbaObjSet>] | |
| | 8BF0 | mov esi,eax | |
| | 68 CC1F4000 | push crackme17-native.401FCC | 401FCC:L"REGISTERED" |
| | 56 | push esi | |
| | 8B0E | mov ecx,dword ptr ds:[esi] | |
| | FF51 54 | call dword ptr ds:[ecx+54] | |
| | 3BC3 | cmp eax,ebx | |
| | DBE2 | fnclx | |
| | 7D 0F | jge crackme17-native.402A45 | |
| | 6A 54 | push 54 | |
| | 68 E41F4000 | push crackme17-native.401FE4 | |
| | 56 | push esi | |

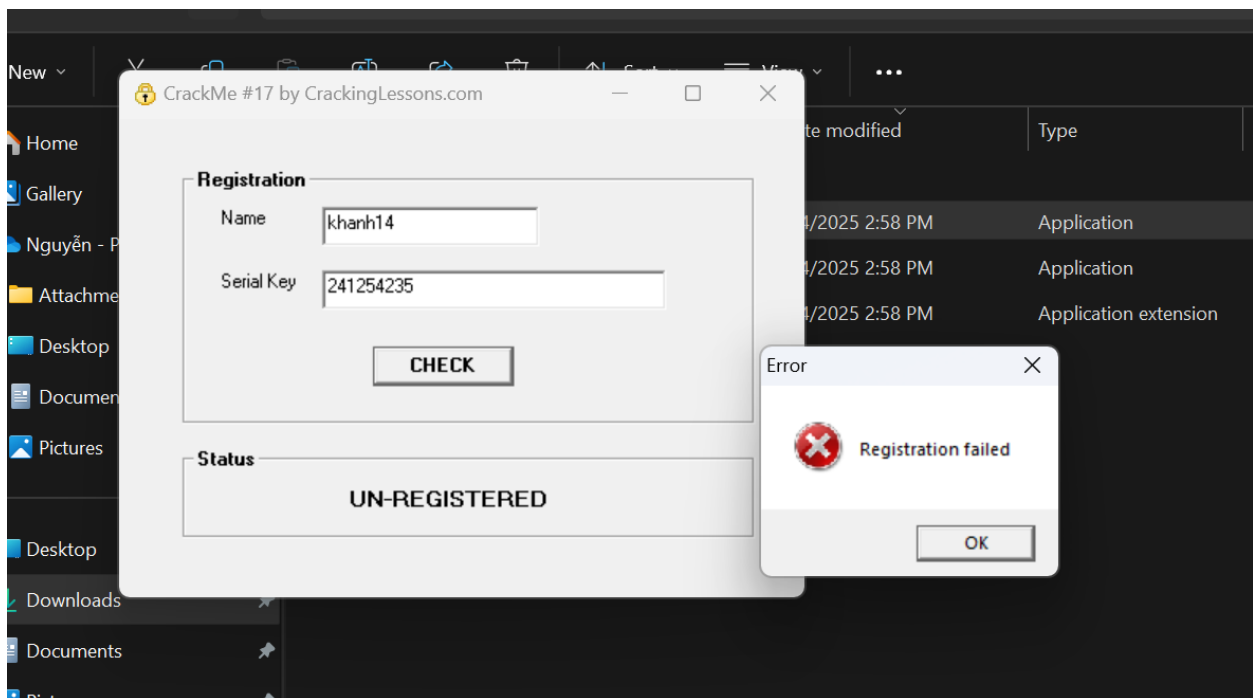
Change this JE command to jump to the next line of code at address 402A0F

| | | | |
|--|----------|---------------|-----------------------------|
| | 00402A03 | 83C4 18 | add esp,18 |
| | 00402A06 | 66:3BFB | cmp di,bx |
| | 00402A09 | 74 04 | je crackme17-native.402A0F |
| | 00402A0B | 90 | nop |
| | 00402A0C | 90 | nop |
| | 00402A0D | 90 | nop |
| | 00402A0E | 90 | nop |
| | 00402A0F | 8B16 | mov edx,dword ptr ds:[esi] |
| | 00402A11 | 56 | push esi |
| | 00402A12 | FF92 14030000 | call dword ptr ds:[edx+314] |
| | 00402A18 | 50 | push eax |

Patch it and check.



2.



Instead of using xdbg tool to view raw assembly code, I using VB decompiler to change pcode to VB structure to easily crack keygen.

This is what I converted to VB source from pcode.exe to see the flow of keygen production:

File Tools Plugins Help

FileName: C:\Users\Lenovo legion 5\Downloads\CrackMe17\CrackMe17\CrackMe17-pcode.exe

P-Code

Decompiler Disassembler HEX Editor

```
Private Sub btnRegister_Click() '402724
    'Data Table: 401FB4
    Dim var_F8 As Integer
    loc_4025BF: var_90 = "49582F35A94345"
    loc_4025D7: var_88 = Me.txtName.Text
    loc_4025F2: var_8C = Me.txtSerialKey.Text
    loc_402602: If (Len(var_88) < 5) Then
        loc_402626: MsgBox("Name should be more than 4 characters", &H10, "Error", var_F8, var_118)
        loc_402636: Exit Sub
    loc_402637: End If
    loc_402661: var_F8 = 4
    loc_402676: var_118 = Mid(var_90, 4, var_F8)
    loc_402691: If (CVar(var_8C) = Ucase(Right(var_88, 4)) & var_118) Then
        loc_4026A1: Me.labelStatus.Caption = "REGISTERED"
        loc_4026CA: MsgBox("Successful registration", &H40, "Congrats", var_F8, var_118)
    loc_4026DD: Else
        loc_4026EA: Me.labelStatus.Caption = "UN-REGISTERED"
        loc_402713: MsgBox("Registration failed", &H10, "Error", var_F8, var_118)
    loc_402723: End If
    loc_402723: Exit Sub
End Sub
```

var_90: include "49582F35A94345" (hardcoded)

```
loc_402661: var_F8 = 4
loc_402676: var_118 = Mid(var_90, 4, var_F8)
```

Mid(var_90, 4, var_F8): take 4 characters from position 4 (F8) -> **82F3**

var_88: include name user typing

```
loc_402602: If (Len(var_88) < 5) Then
    loc_402626: MsgBox("Name should be more than 4 characters", &H10, "Error", var_F8, var_118)
    loc_402636: Exit Sub
loc_402637: End If
```

condition: at least 5 characters, if not met the requirement, displaying error and exit.

var_8c: include serial key using typing (expected key value)

```
loc_402691: If (CVar(var_8C) = Ucase(Right(var_88, 4)) & var_118) Then
    loc_4026A1: Me.labelStatus.Caption = "REGISTERED"
    loc_4026CA: MsgBox("Successful registration", &H40, "Congrats", var_F8, var_118)
loc_4026DD: Else
    loc_4026EA: Me.labelStatus.Caption = "UN-REGISTERED"
    loc_402713: MsgBox("Registration failed", &H10, "Error", var_F8, var_118)
loc_402723: End If
loc_402723: Exit Sub
```

If Serial = Right(Name, 4).ToUpper() + "82F3" => we register successfully

If not => Un-Registered status

Example:

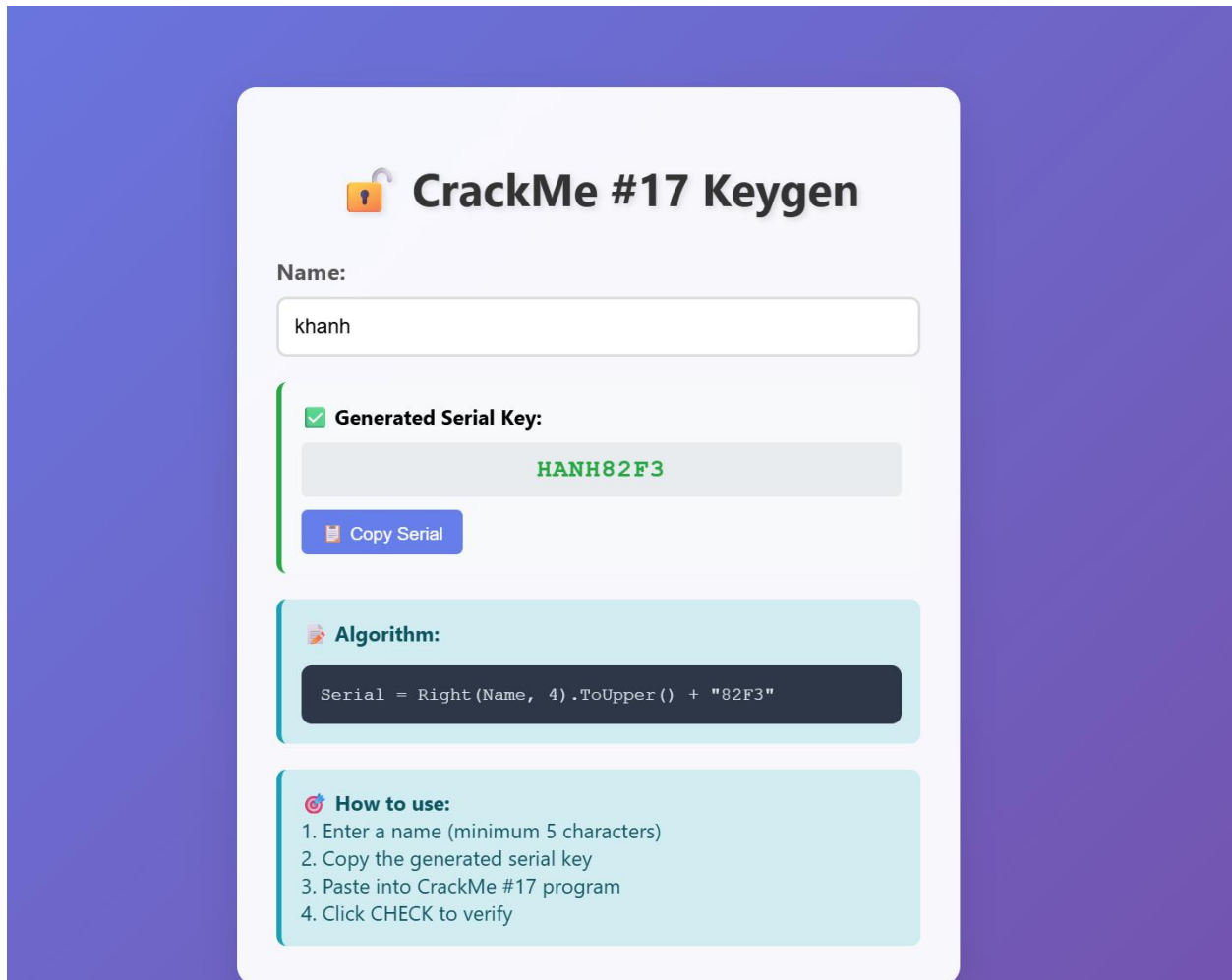
Name: "khanh"

Right(Name, 4): "min" (last 4 characters)

Ucase(): "HANH" (convert to uppercase)

Serial: "HANH" + "82F3" = "**HANH82F3**"

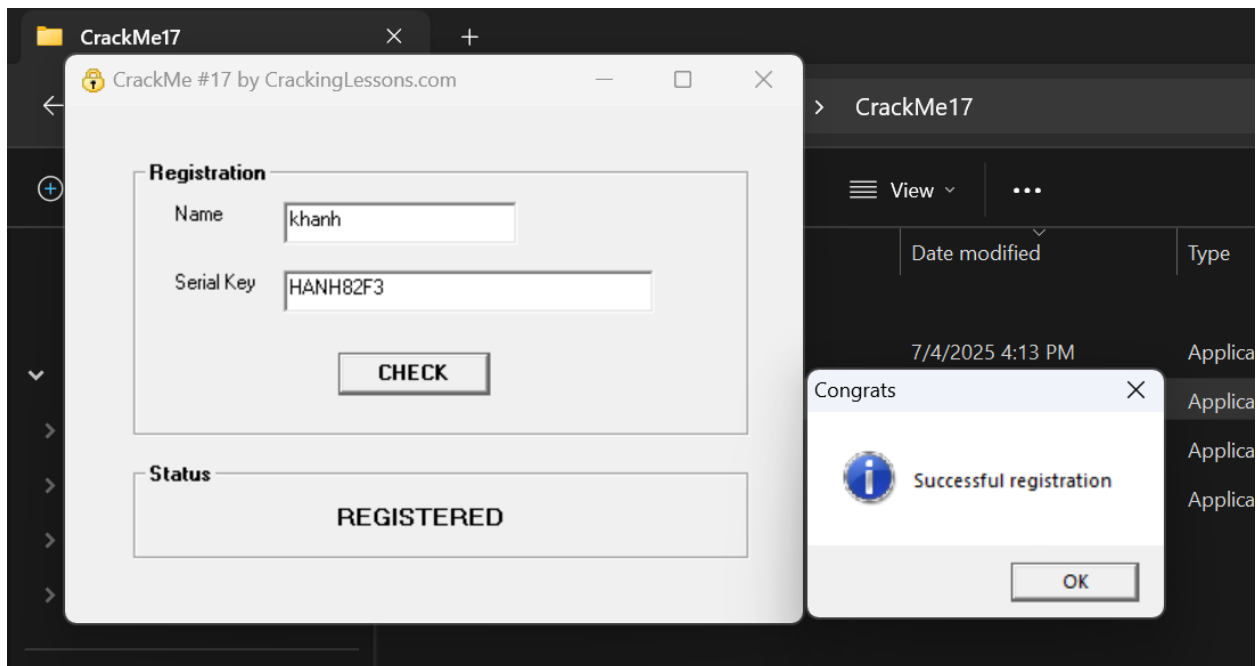
For this challenge, instead of patch file to display keygen, I create a js program to find keygen after typing name:



The image shows a web application titled "CrackMe #17 Keygen" with a lock icon. It features a "Name:" input field containing the text "khanh". Below the input field, there is a green checkmark icon and the text "Generated Serial Key:". The generated serial key "HANH82F3" is displayed in green text on a light gray background. A blue button with a document icon and the text "Copy Serial" is located below the serial key. Further down, there is a section titled "Algorithm:" with a code block containing the JavaScript code: `Serial = Right (Name, 4) .ToUpper () + "82F3"`. At the bottom, there is a section titled "How to use:" with a list of four steps: 1. Enter a name (minimum 5 characters), 2. Copy the generated serial key, 3. Paste into CrackMe #17 program, and 4. Click CHECK to verify.

<https://claude.ai/public/artifacts/d38a7713-3189-4a7b-970b-95c17c83d72f>

NGUYEN NAM KHANH – HE191159 – IA1902 – IAM302



DONE!!!