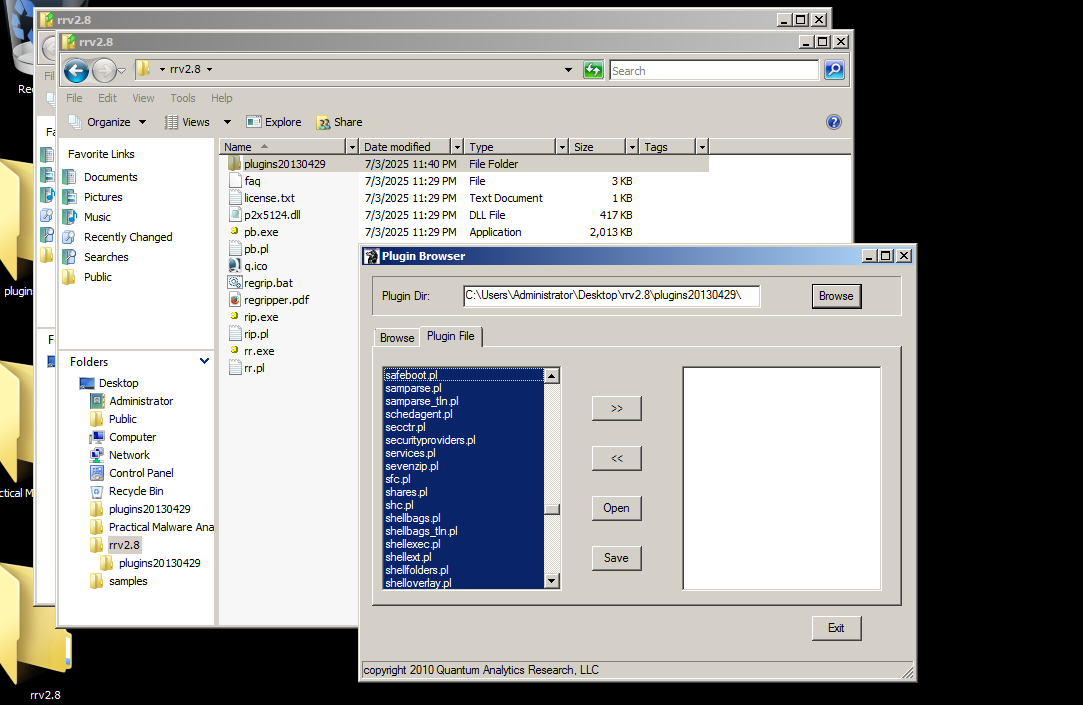
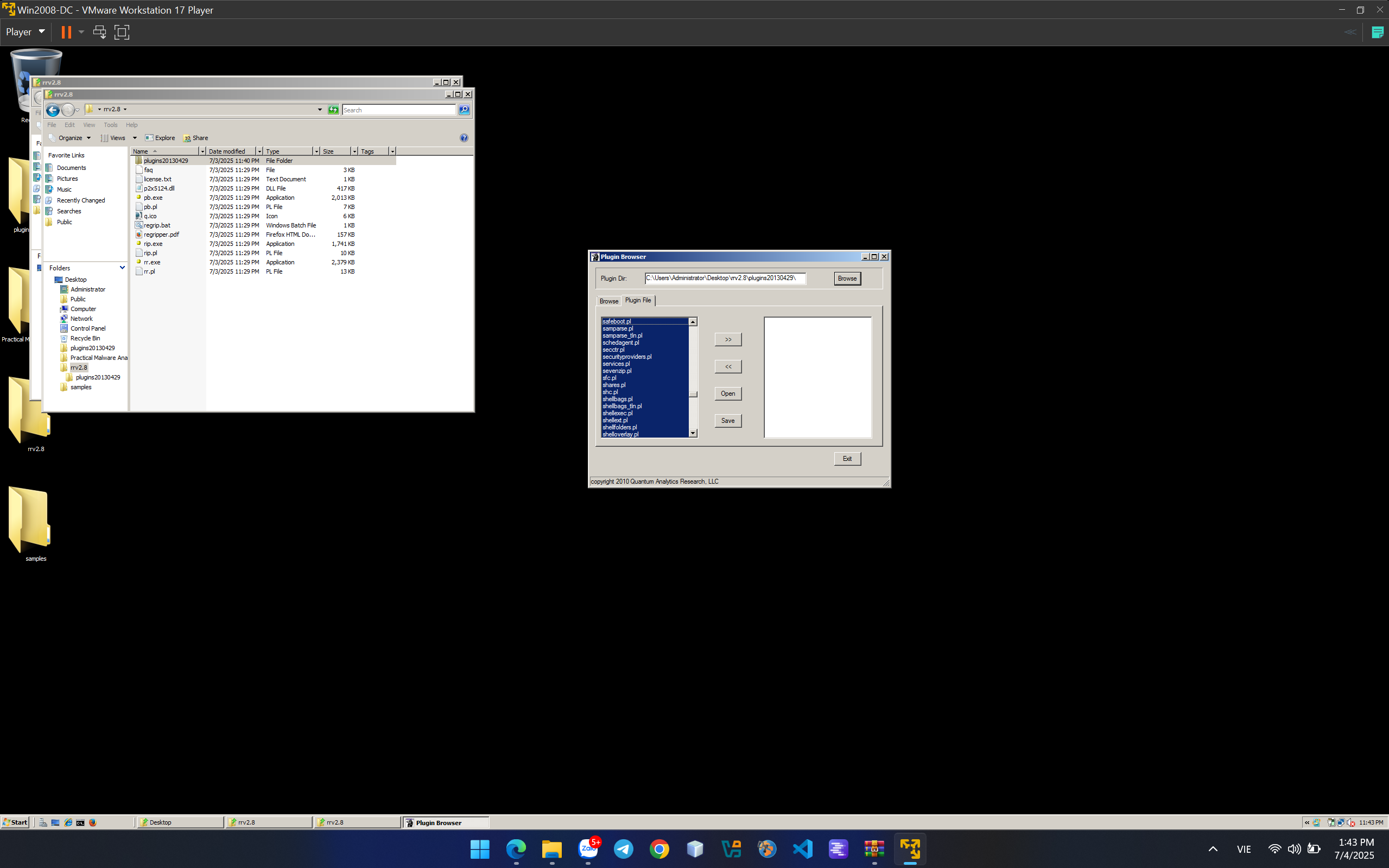
**LAB16: Registry Forensics with RegRipperPlug-ins**

**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

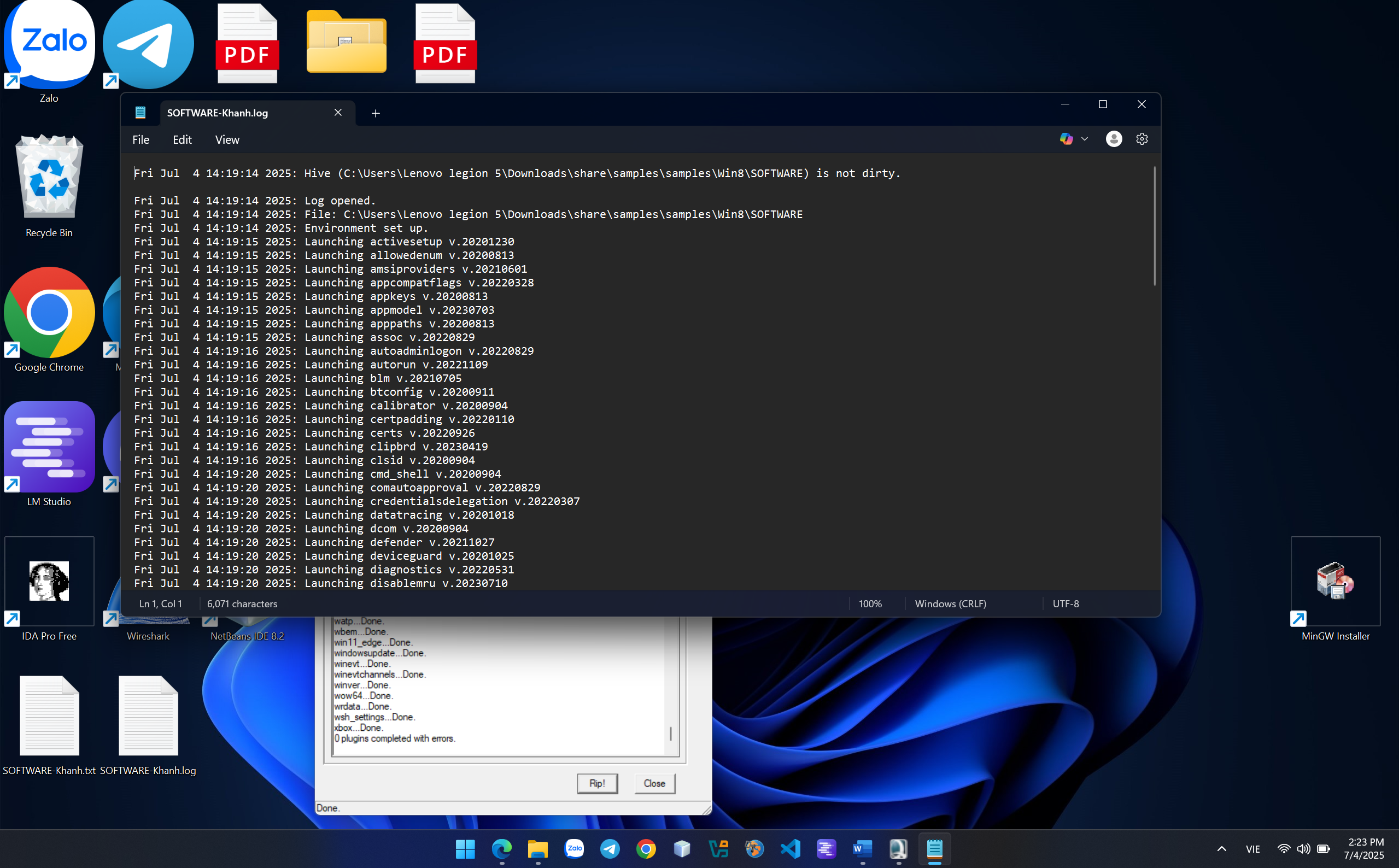
- Method 2: In the file rrv2.8 there is a program named "pb" and there it will display the interface window so that you can import the plugin's files.

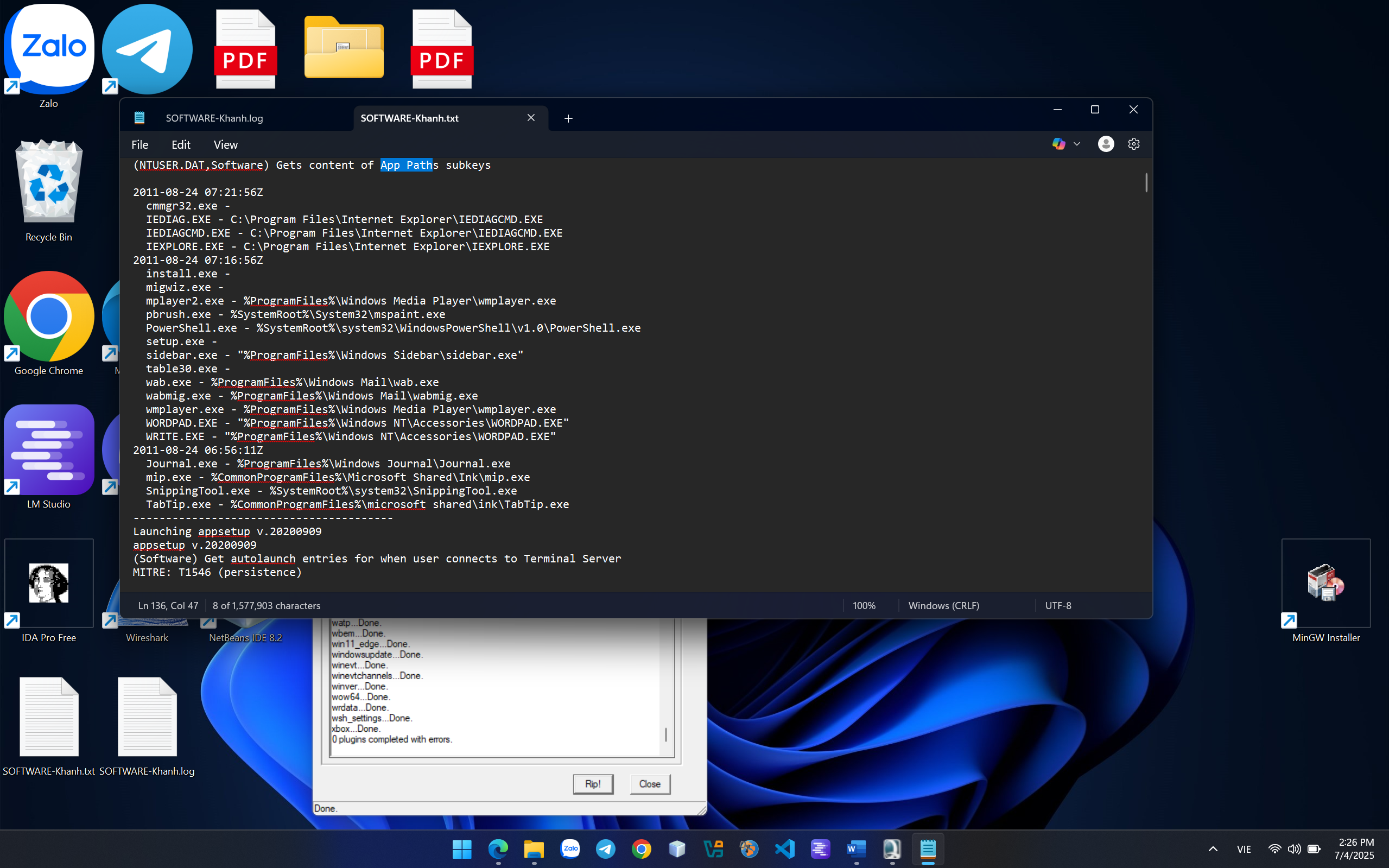
Unfortunately, after using vmware win 2008 and import plugin file to regripper 2.8 unsuccessfully, I change to my real computer and update version to 4.0 and it import automatically plugin file.

And here we will conduct the analysis:

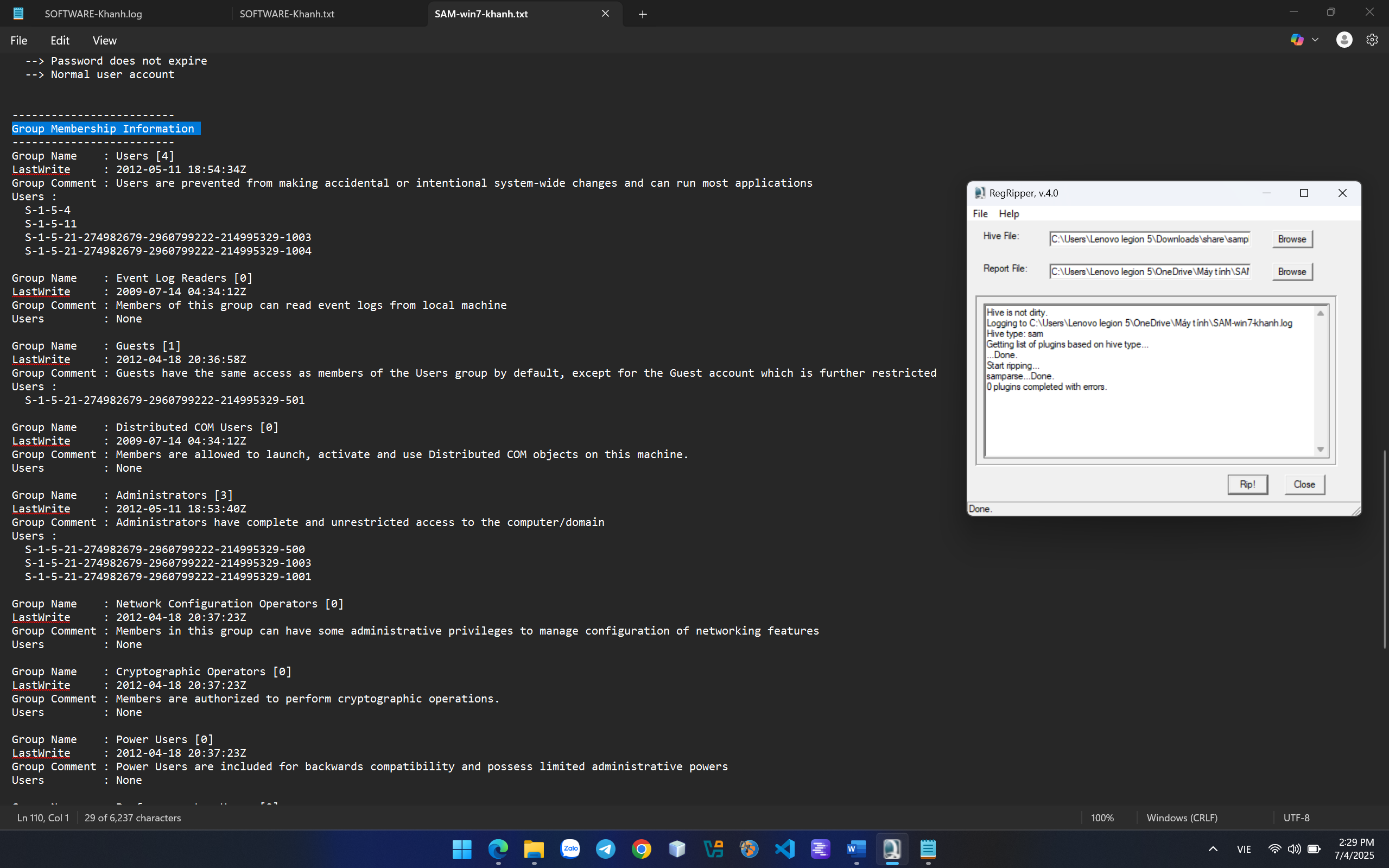
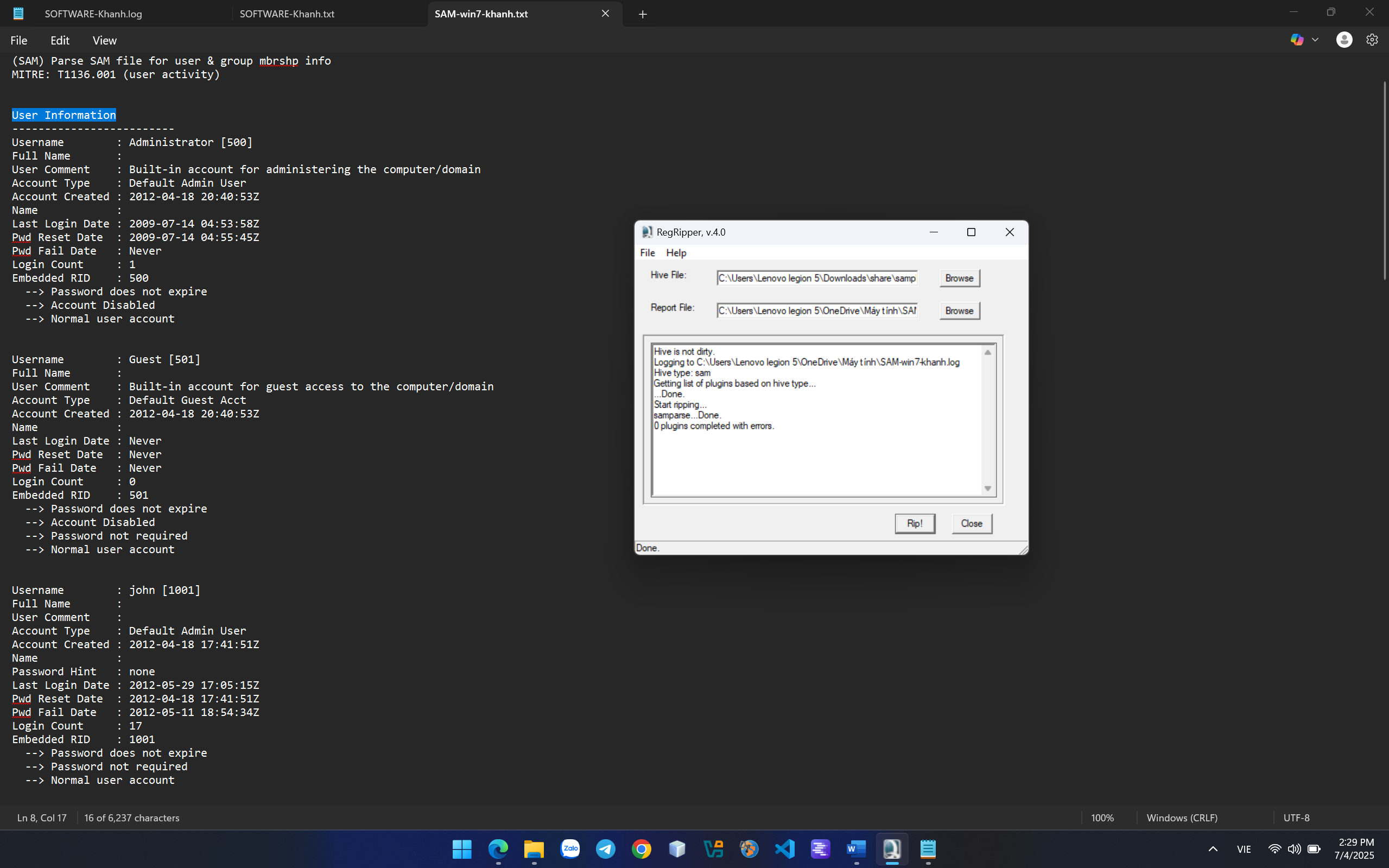
**Analyst file Software:**



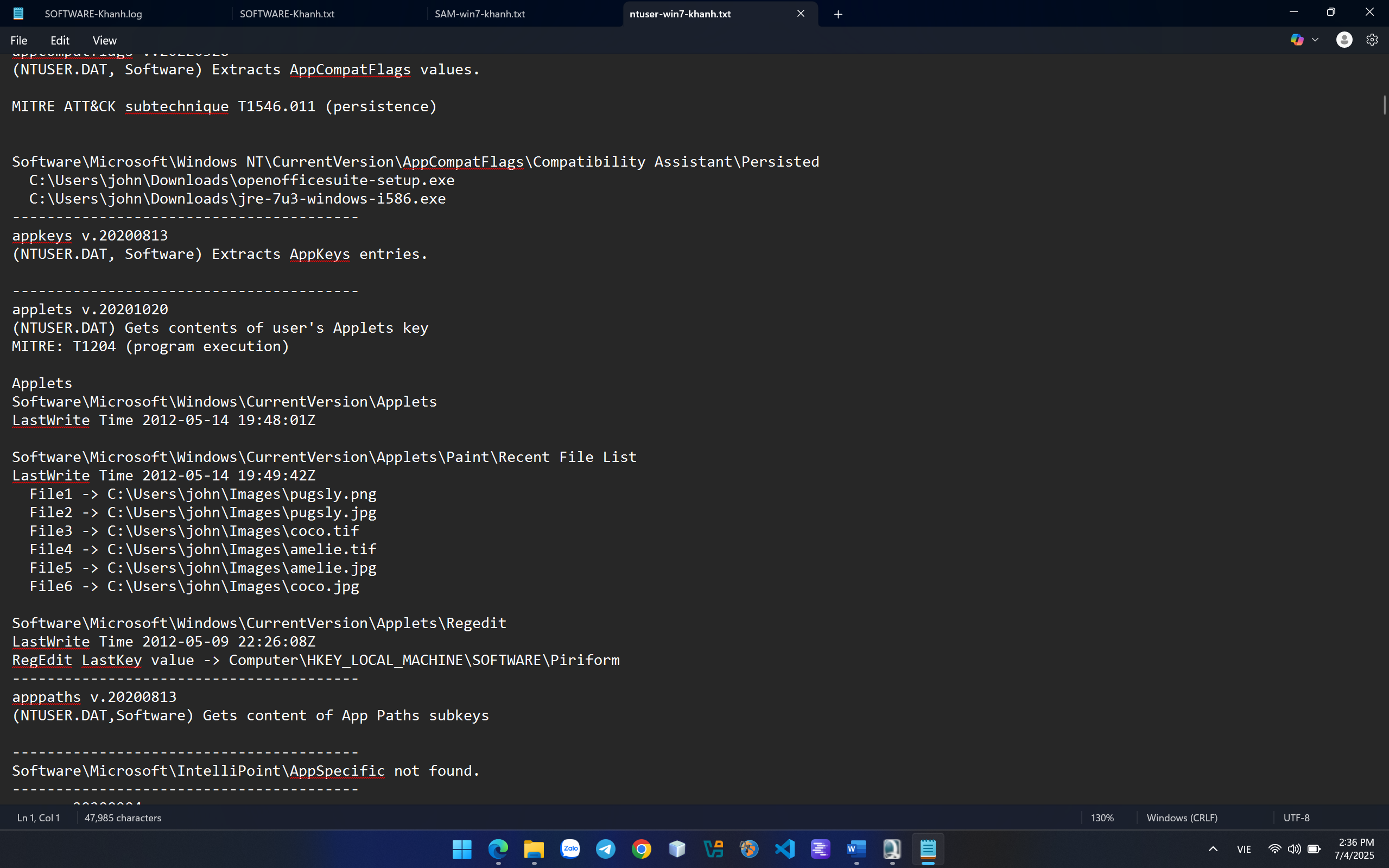
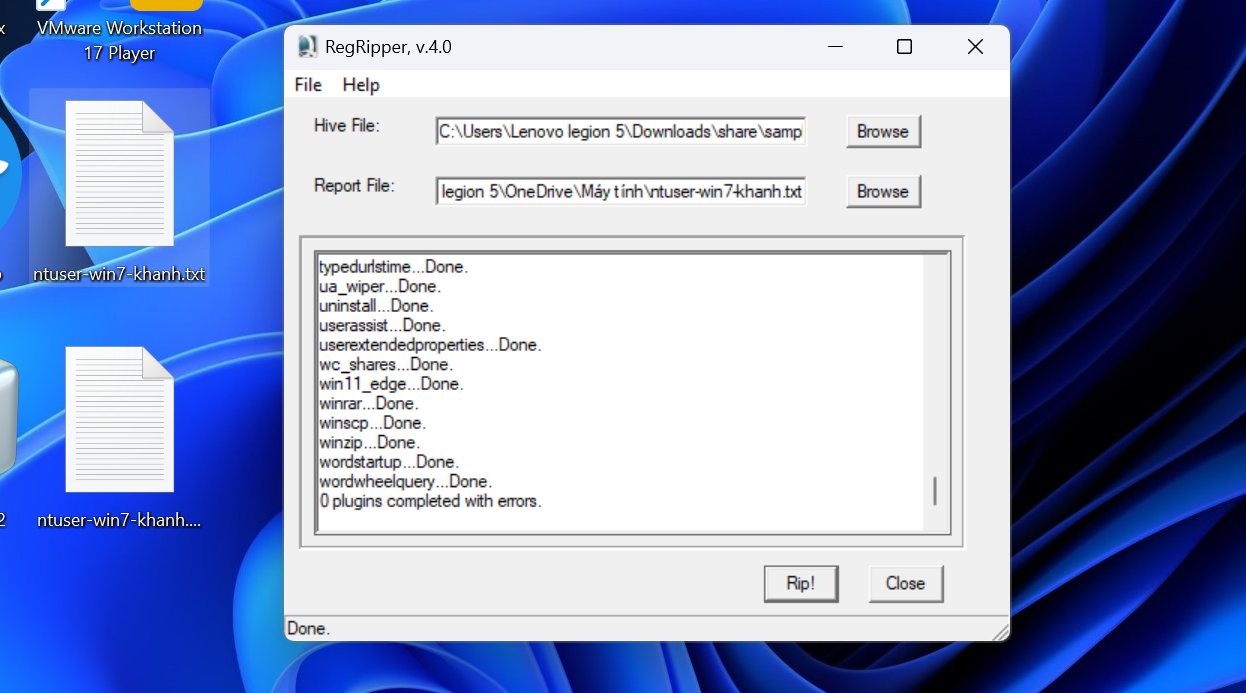
File log: 

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, inforamtion, etc …

**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:**

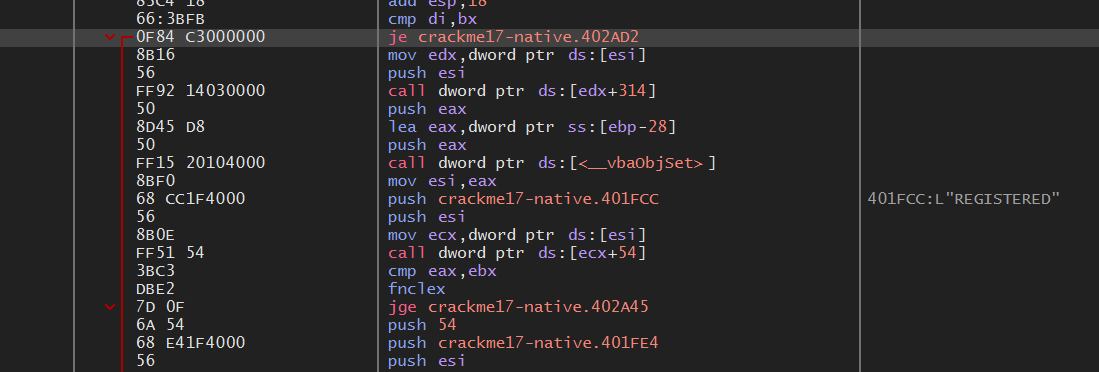
This crackme is written in Visual Basic 6 and compiled as a p-code executable.

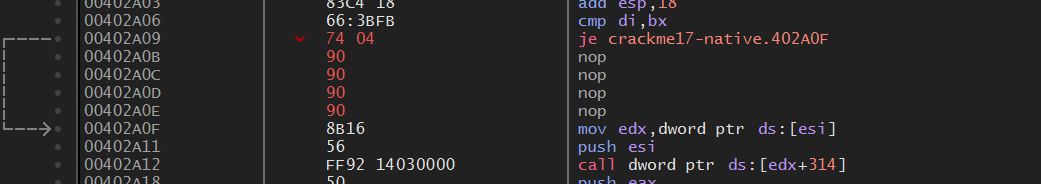
The objectives of this crackme are:

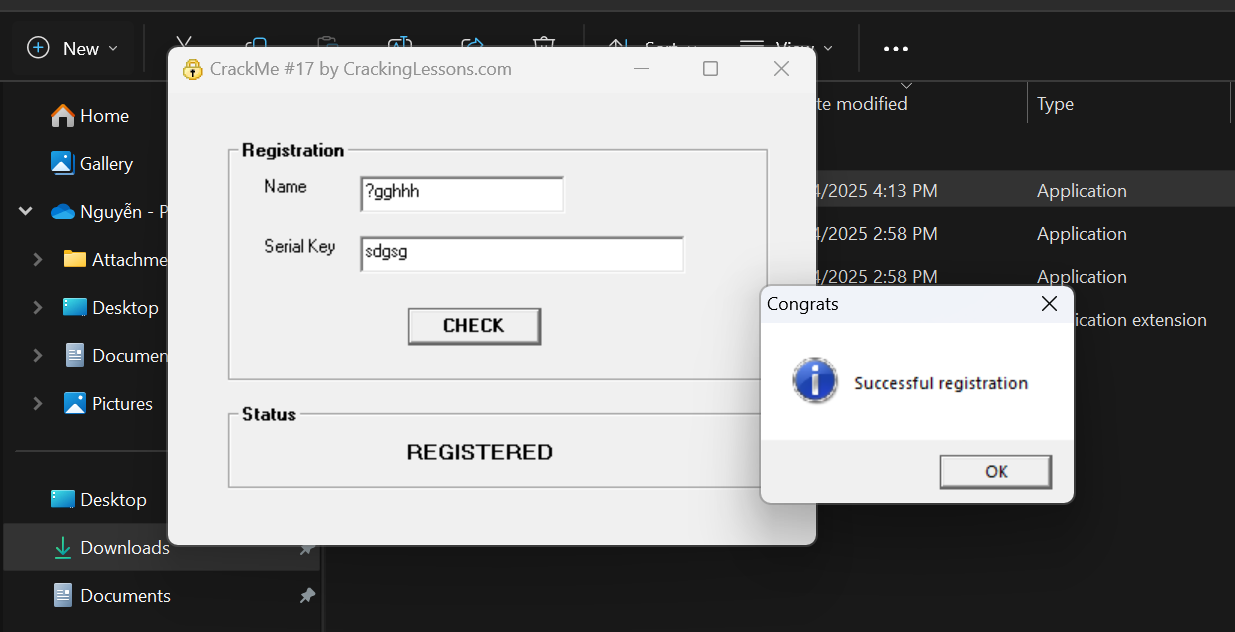
1. patch the file so that no matter what name or serial key you enter, it will become registered
2. create a keygen for it

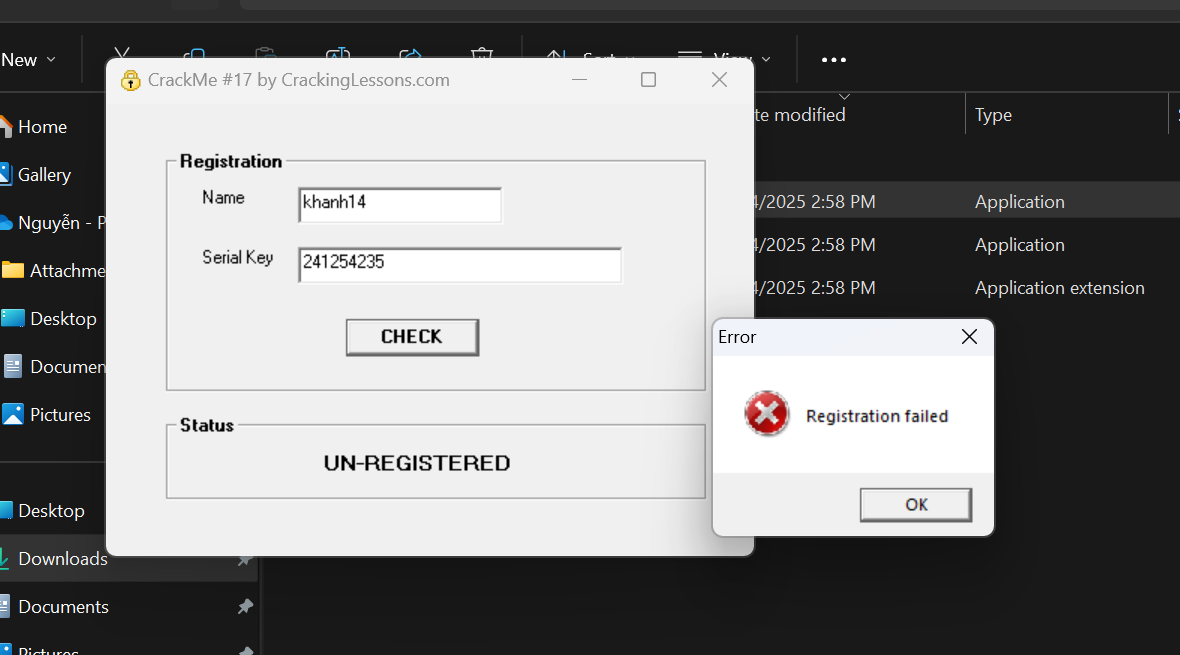
**1.**

Find string reference of REGISTERED status

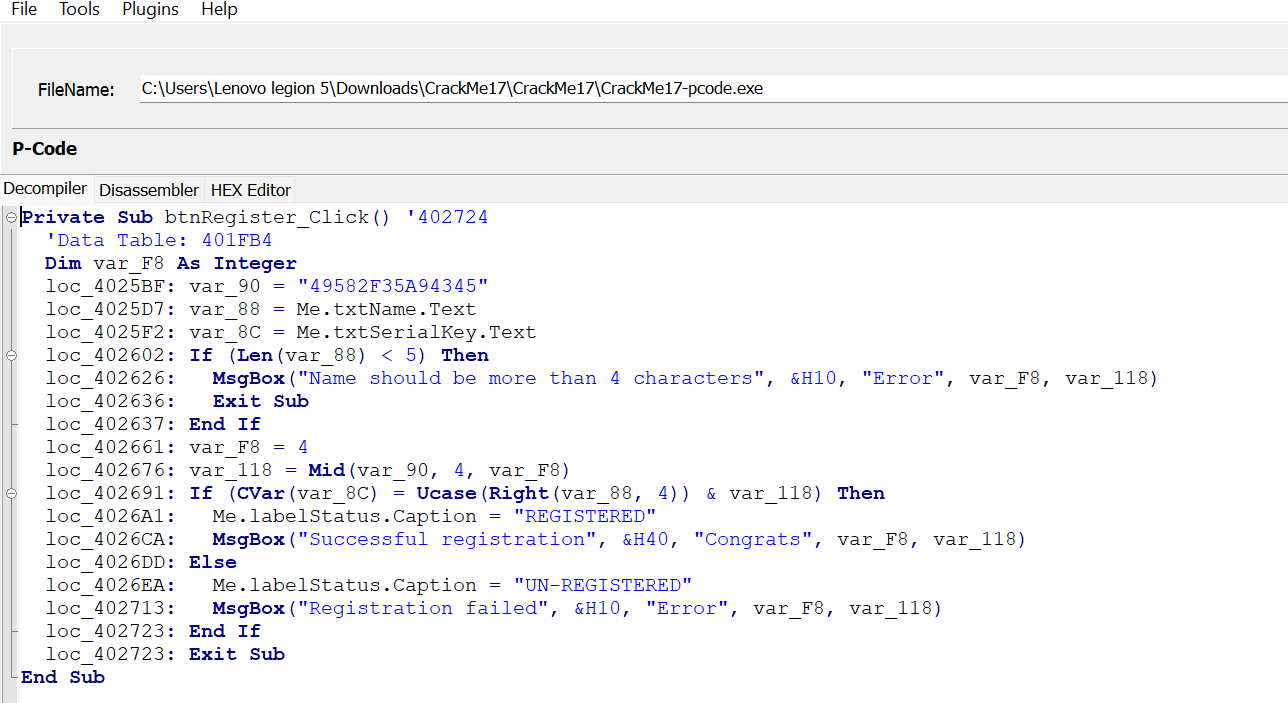
Found that there is a JE jump through REGISTERED status to UN-REGISTERED

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

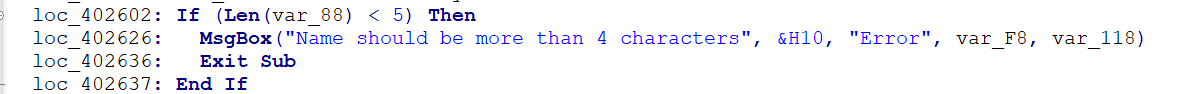
Patch it and check.

**2.**

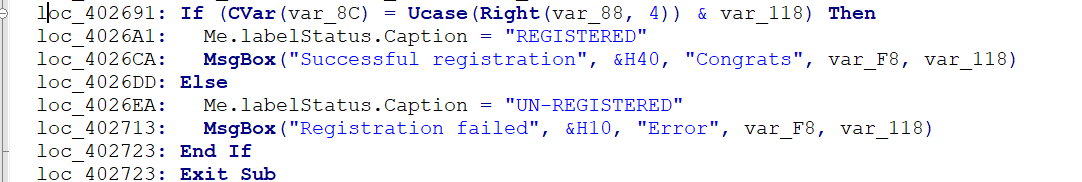
Instead of using xdbg tool to view raw assembly code, I using VB decompiler to change pcode to VB structure to easily crack keygen.

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

**var\_90**: include "49582F35A94345" (hardcoded) **Mid(var\_90, 4, var\_F8)**: take 4 characters from position 4 (F8) -> **82F3**

**var\_88**: include name user typingcondition: at least 5 characters, if not met the requirement, displayling error and exit.

**var\_8c**: include serial key using typing (expected key value)



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

If not => Un-Registered status

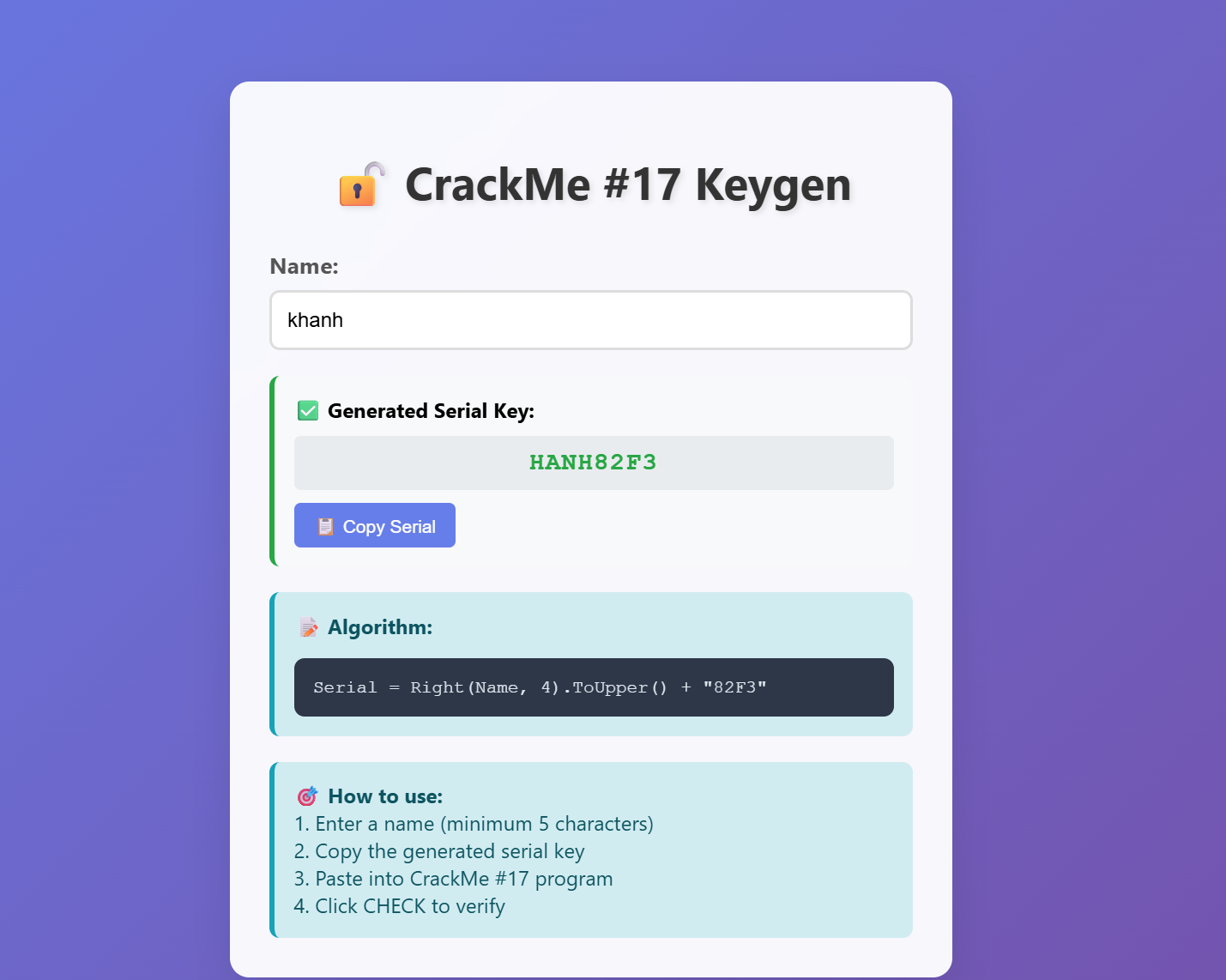
Example:

**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: 

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



DONE!!!