



Instituto Tecnológico de San Juan del Río



Tópicos de ciberseguridad

P R E S E N T A:

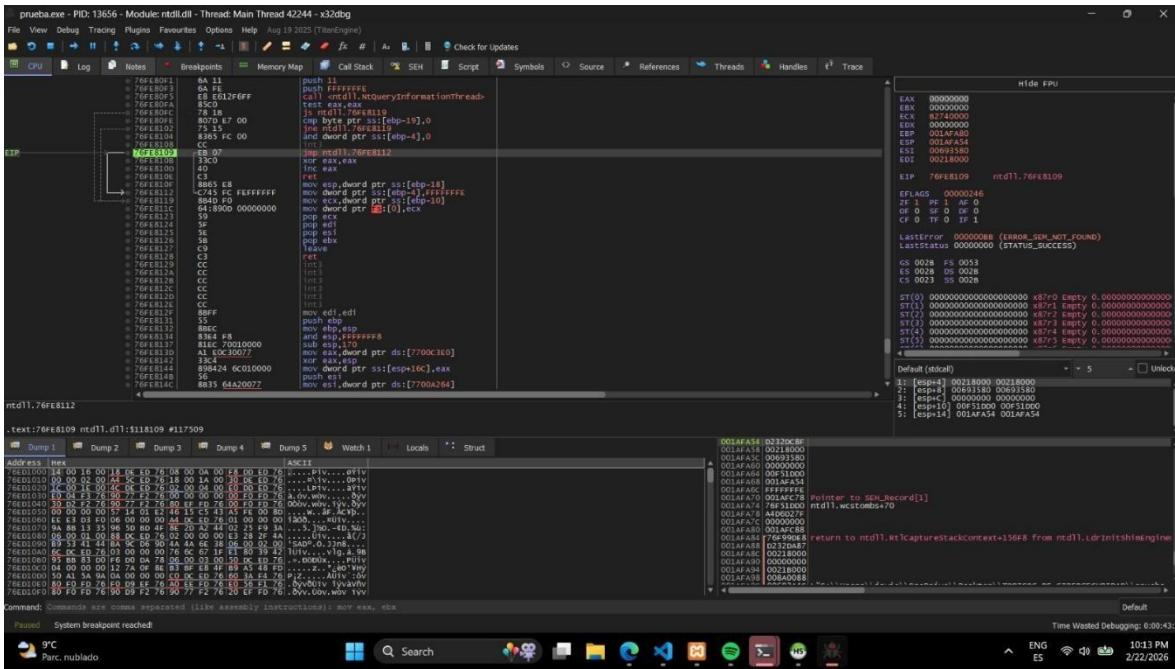
Pablo Colunga Ochoa

Andres Martin Sinecio

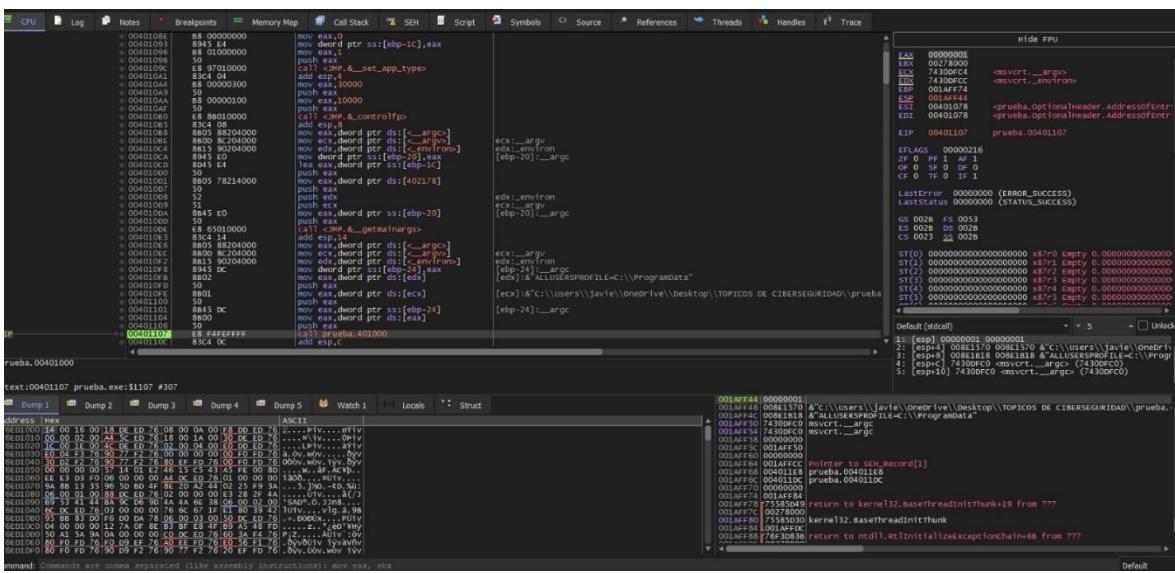
Javier Hernandez Mendoza

Ingeniería en Sistemas Computacionales

PERIODO [ENE-JUN 2026]



cargamos el archivo "prueba" dentro del debuggerx32



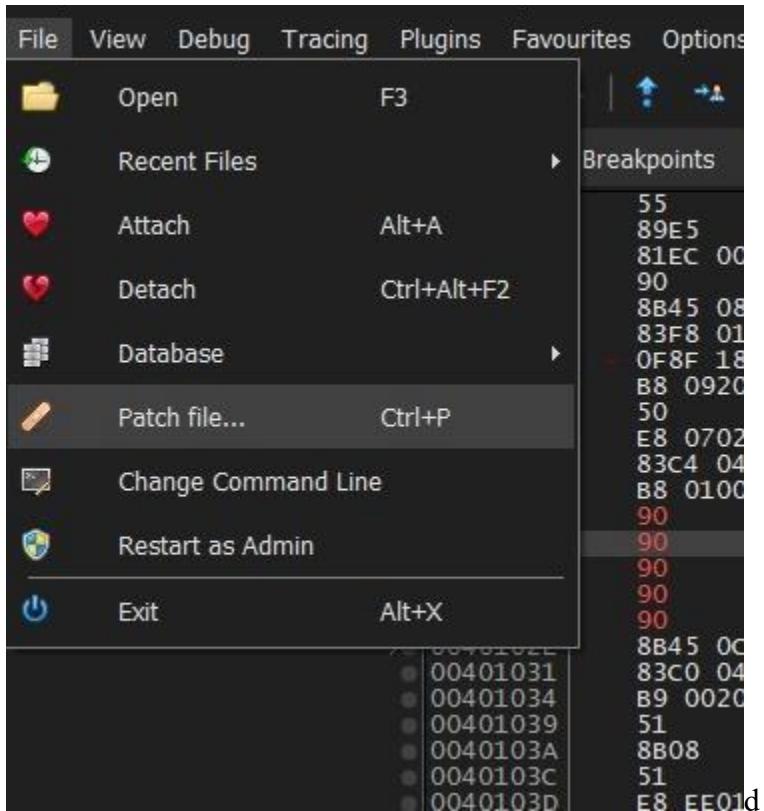
Identifiqué el main porque después de __getmainargs (que construye argc/argv), el runtime hace call prueba.00401000, y al hacer Step Into se entra al código propio del ejecutable, que corresponde al inicio de la lógica principal.

Después de la inicialización del runtime (MSVCRT), al ejecutar Step Into sobre call prueba.00401000, el flujo entra a prueba.exe en la dirección 00401000. En esta función ya se observa lógica propia del programa: lectura de argumentos (argv), comparación de cadenas y mensajes de salida, lo que indica que corresponde al inicio de la rutina principal (main lógico).

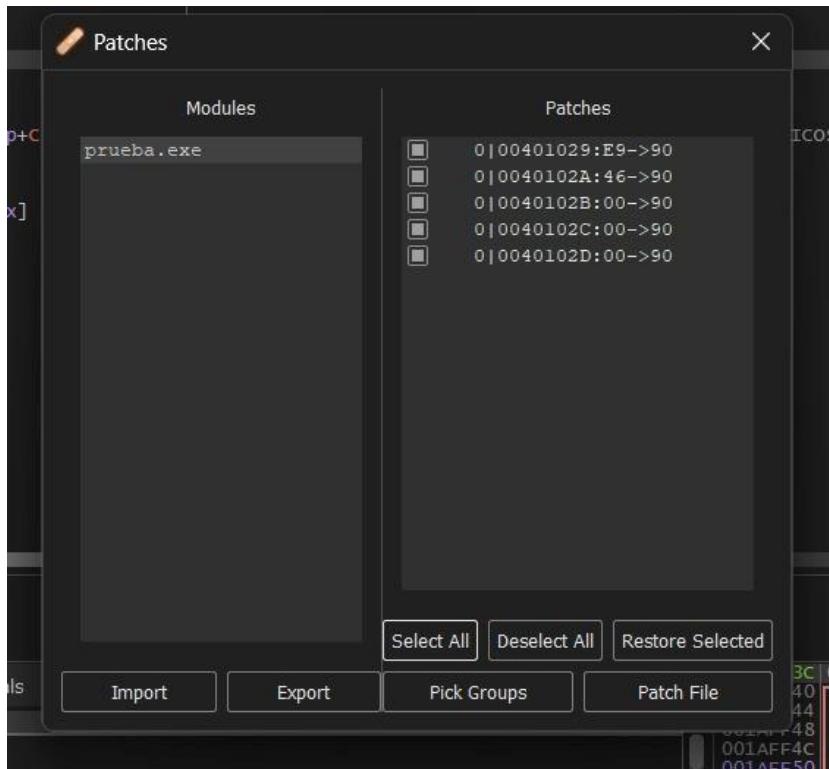
The screenshot shows the Immunity Debugger interface with the following details:

- Registers:** CPU tab selected. Registers show values like EAX=00000000, EBX=0029A000, ECX=74000000, etc.
- Stack:** Stack dump shows memory starting at 00401000, containing assembly instructions and some strings like "push esp", "add esp, 4", and "nop".
- Memory Map:** Shows the program's memory layout with sections like ".text", ".data", ".rsrc", and ".rsrc\$".
- Call Stack:** Call stack shows the history of function calls.
- Script:** Script tab is visible.
- Symbols:** Symbols tab is visible.
- Source:** Source tab is visible.
- References:** References tab is visible.
- Threads:** Threads tab is visible.
- Handles:** Handles tab is visible.
- Trace:** Trace tab is visible.
- Registers:** Registers tab is visible.
- Registers pane:** Shows registers EAX through EDI, their current values, and their previous values.
- Registers pane controls:** Includes buttons for "Keep Size", "Fill with NOPs", "XORParse", and "asm/jit".
- Registers pane status:** "Instruction encoded successfully! Bytes: 90"
- Registers pane options:** "optionalHeader.AddressOfEntryToInit" is selected.
- Registers pane dropdown:** Shows options for "Default (stdcall)" and "Unlock".
- Registers pane assembly:** Displays assembly code for the current instruction, including pushes to the stack and jumps to "prueba_0401029".
- Registers pane stack dump:** Shows the stack dump from address 00401000 to 0040107F.

Se identificó una instrucción de transferencia de control (salto) en el flujo principal de ejecución y se sustituyó por instrucciones NOP, eliminando la redirección y forzando la continuidad secuencial del código. La modificación se realizó mediante re-ensamblado en x32dbg, quedando registrada en la dirección virtual correspondiente y reflejada en el byte de operación 0x90, como evidencia del parche aplicado.



Se parcheo el archivo



```
PS C:\Users\javie\OneDrive\Desktop\TOPICOS DE CIBERSEGURIDAD> .\pruebabapycrack.exe hola  
Acceso Concedido  
PS C:\Users\javie\OneDrive\Desktop\TOPICOS DE CIBERSEGURIDAD> |
```

corrimos el programa con cualquier contraseña dandonos el acceso

```
PS C:\Users\javie\OneDrive\Desktop\TOPICOS DE CIBERSEGURIDAD> python .\pycrack.py  
Bytes a modificar: 0f8513000000  
Su parche fue aplicado correctamente> C:\Users\javie\OneDrive\Desktop\TOPICOS DE CIBERSEGURIDAD\pruebaa.exe  
PS C:\Users\javie\OneDrive\Desktop\TOPICOS DE CIBERSEGURIDAD> |
```

se utilizo el python para hacer la modificacion de los bytes correspondientes y se aplica el parche en un archivo distinto al original