Instituto Tecnológico de San Juan del Río

Tópicos de Ciberseguridad

P R E S E N T A:

Maqueda Durazno Alexis

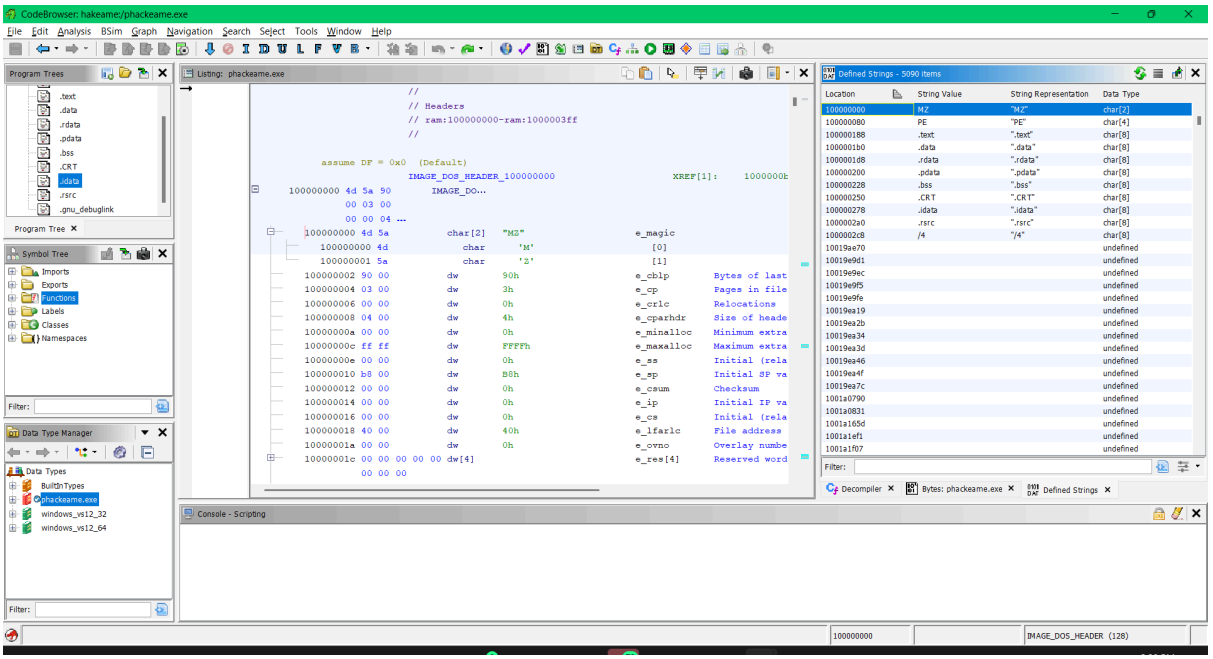
Mendoza Garcia Daniela

Rengel Olvera Paloma

Ing. Sistemas Computacionales

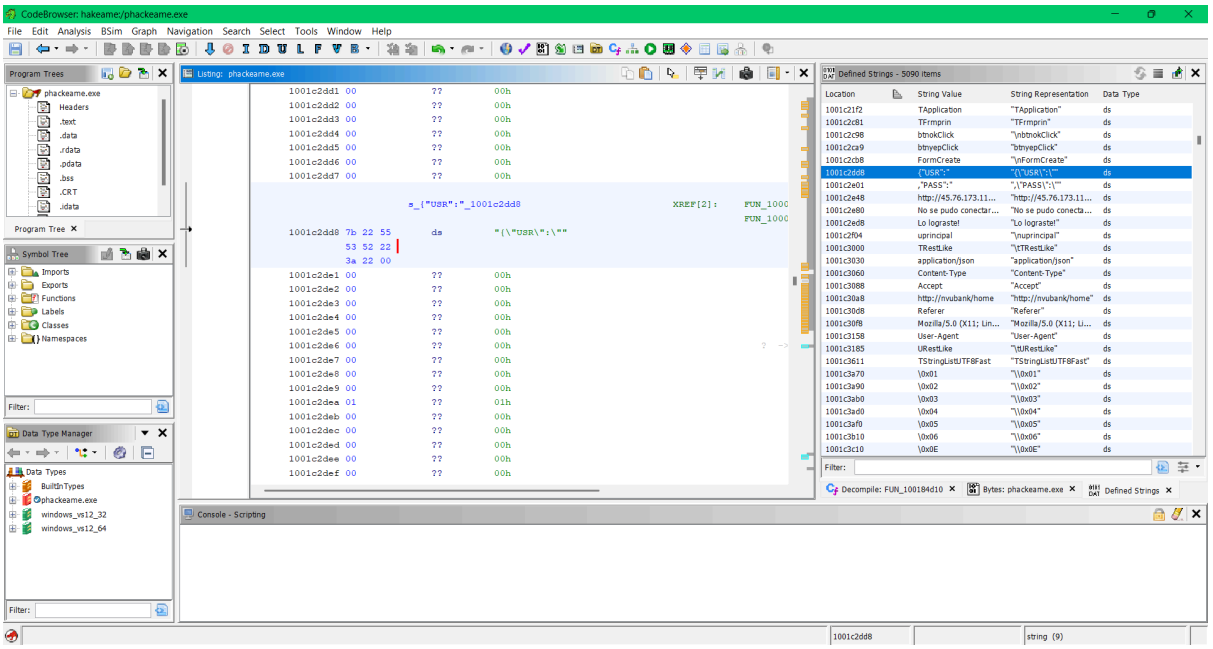
Video de Youtube: https://youtu.be/E9dX-bS26Ww

1- Para comenzar, se requiere iniciar la herramienta de ingeniería inversa Ghidra. El primer paso es crear un nuevo proyecto dentro de Ghidra. Una vez creado el espacio de trabajo, se debe cargar el archivo ejecutable (`.exe`) que se desea analizar. Ghidra importará el binario y comenzará el proceso de análisis inicial, identificando funciones, cadenas de texto y referencias.

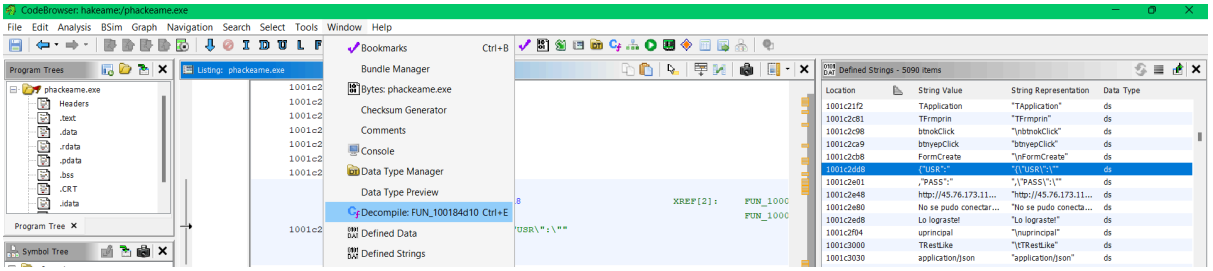


2- Con el ejecutable cargado y analizado, el siguiente paso es la búsqueda de cadenas de texto específicas. Se debe utilizar la ventana o el panel de "Defined Strings". En este panel, se aplicará el buscador (filtro) para localizar cualquier cadena que contenga la subcadena "USR". La localización de estas cadenas a menudo apunta a secciones del código que

manejan la entrada o validación de credenciales de usuario.

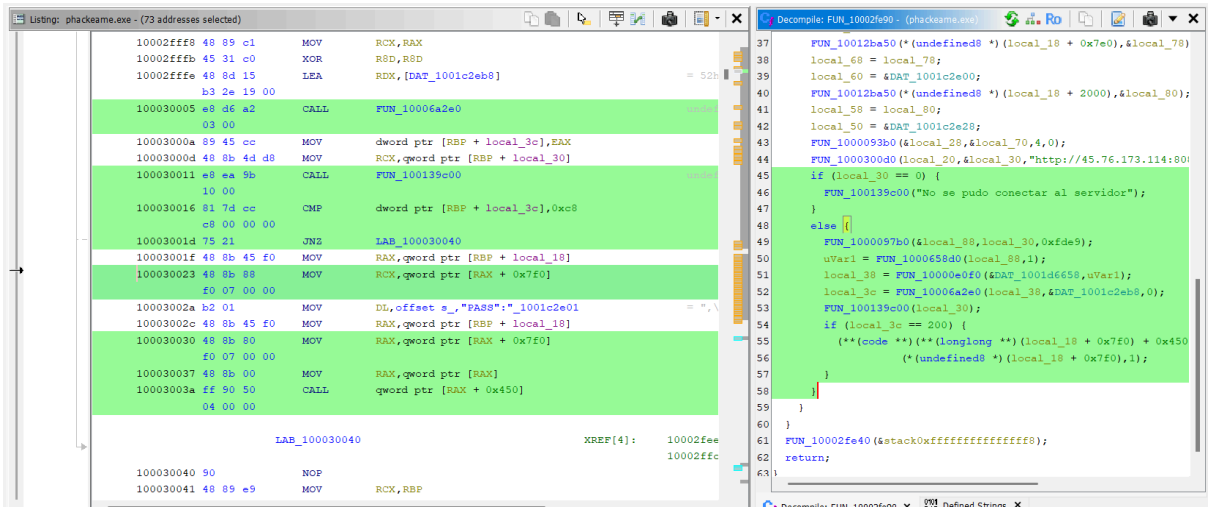


3- Una vez localizadas las referencias a las cadenas de texto en el código ensamblador, se procede a obtener una vista de alto nivel del código. Estando posicionado en las líneas de código relevantes, se accede al apartado del menú "Window" y luego se selecciona "Decompile" (Descompilar).

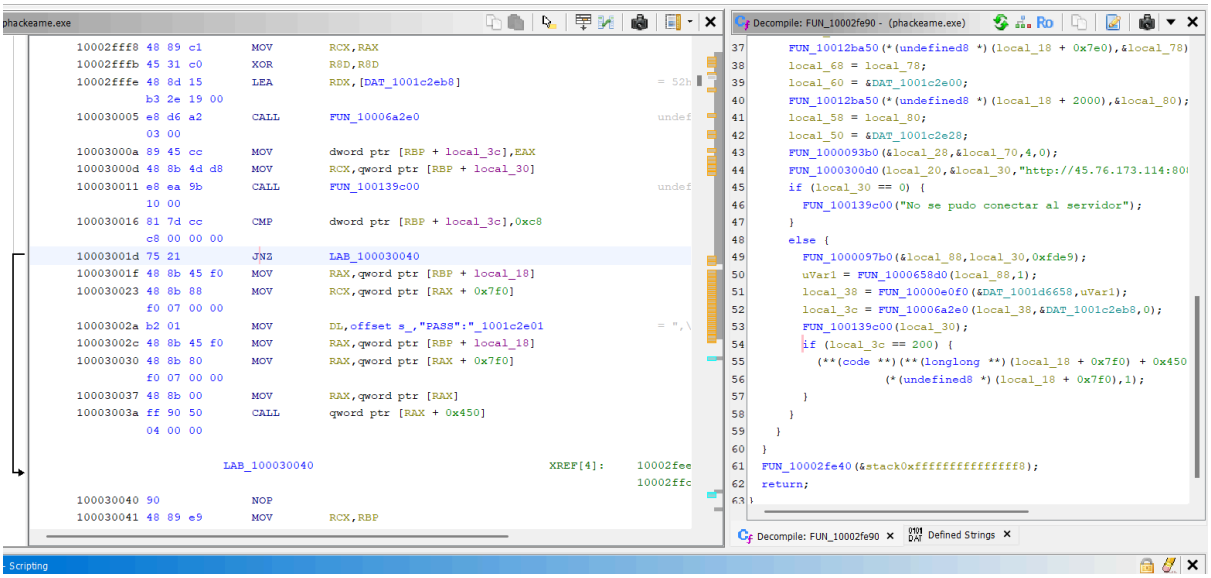


4- La acción anterior resultará en la apertura de la ventana del descompilador, la cual arrojará una aproximación del código fuente base (en un pseudocódigo similar a C/C++) a partir del código ensamblador del `.exe`. Aunque no es el código fuente original exacto, esta

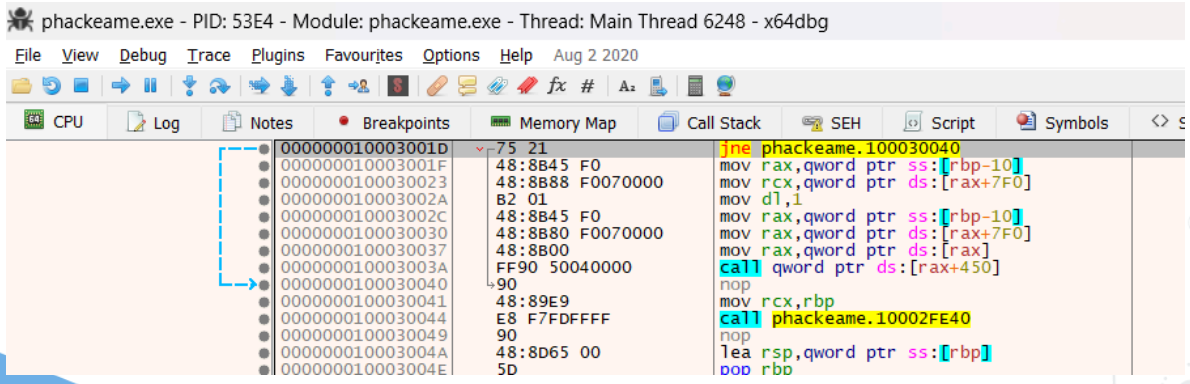
aproximación es invaluable para entender la lógica del programa.

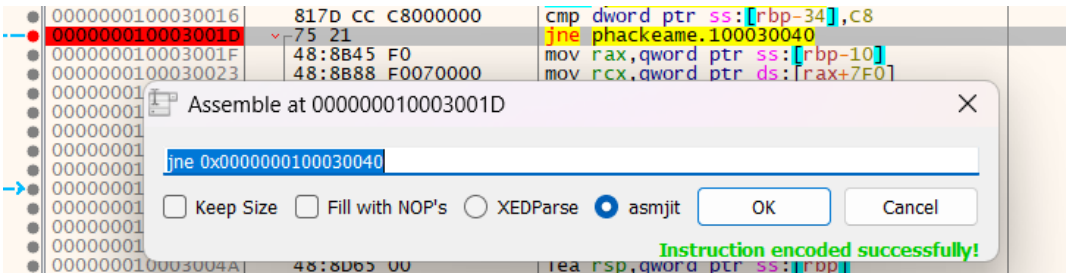


5- Localizamos el If que valida si accedemos o nos deniega la entrada (su posición exaxta, hexadecimal, etc

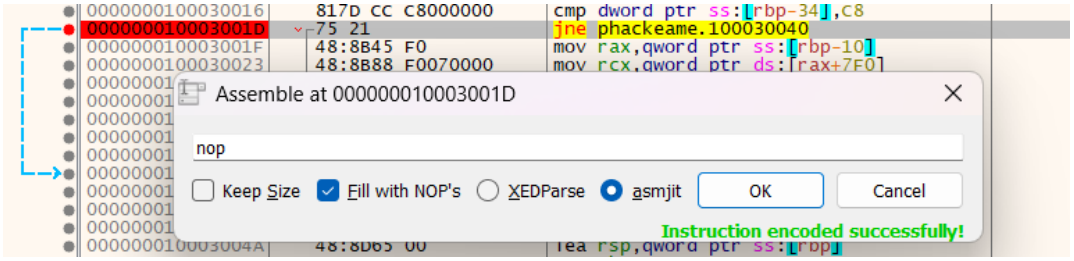


5- entramos a x64dbg y cargamos el .exe, Buscamos la dirección exacta (10003001d) que nos dio Ghidra para ver el comportamiento del procesador en tiempo real.

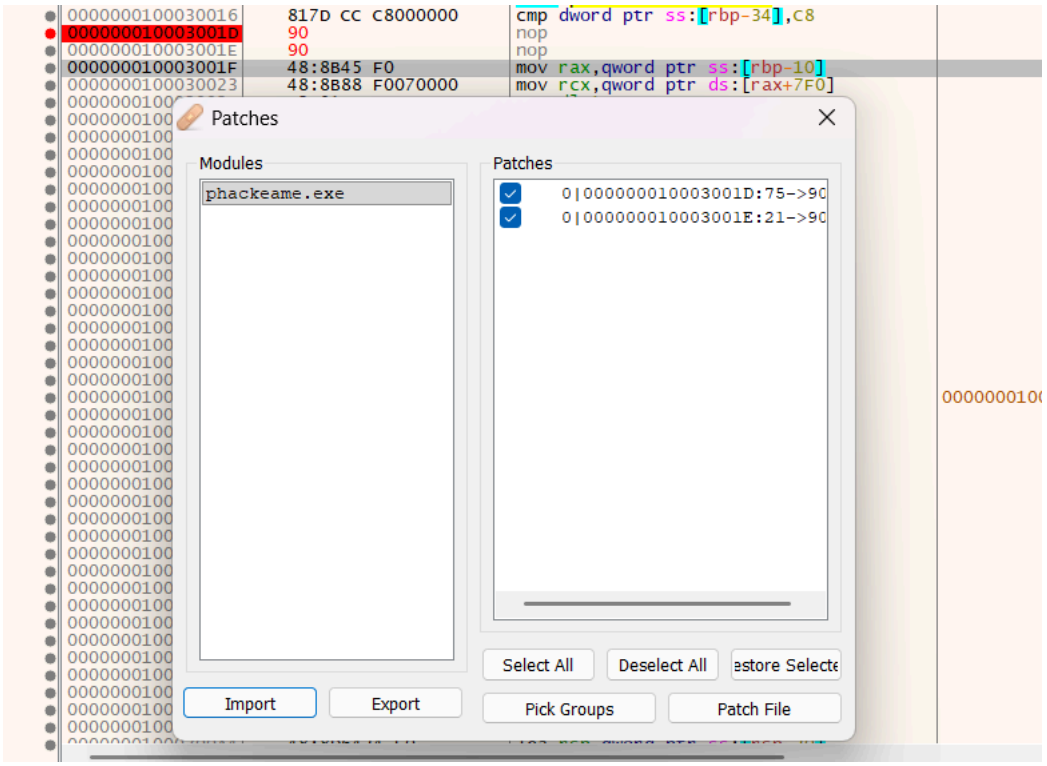


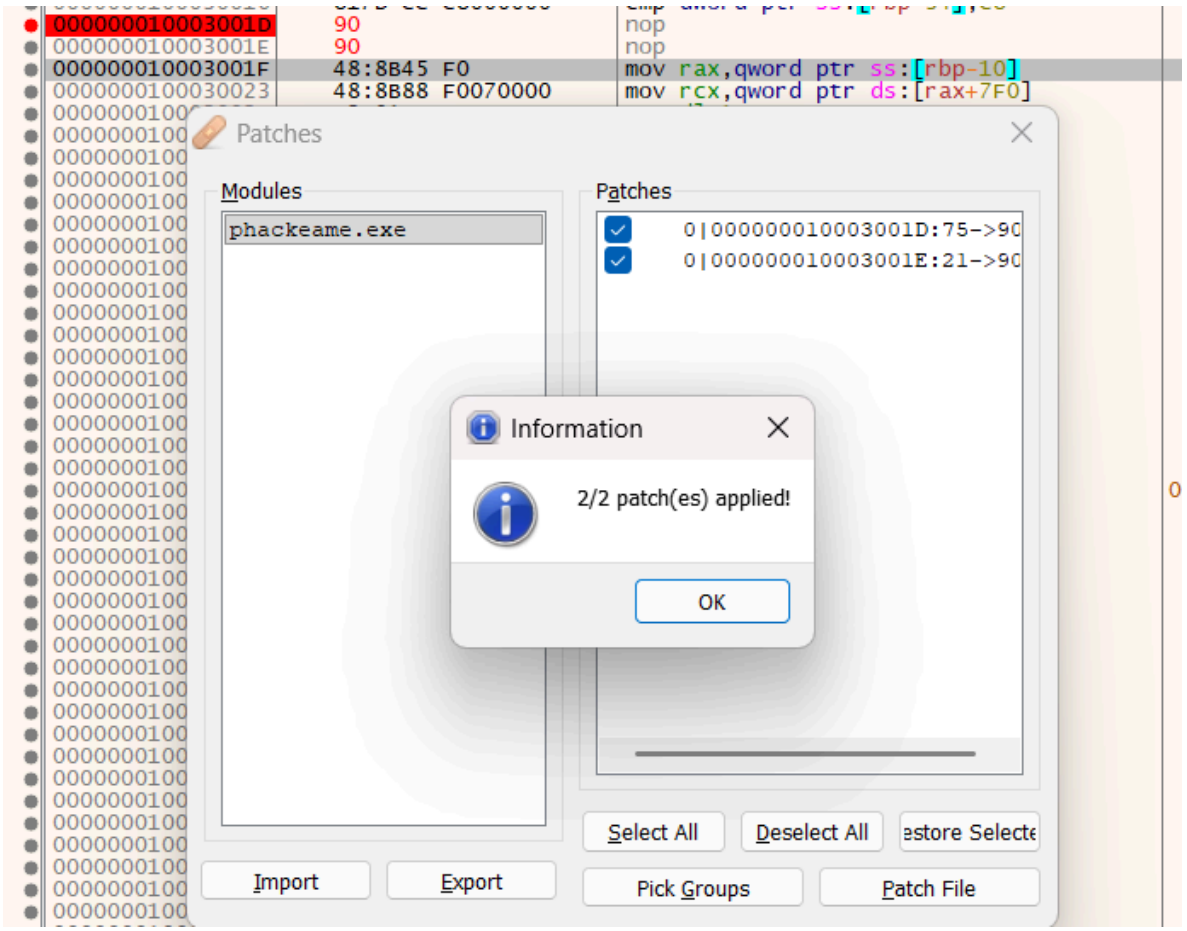


6- Seleccionamos la instrucción de salto (`jne`) y la reemplazamos por instrucciones `nop`. Al poner `nop`, el procesador simplemente "pasa de largo". Ya no importa qué contraseña pongas; el programa nunca saltará al error y siempre te dará acceso.



7- Aplicamos los parches (`2/2 patches applied`) para que los cambios sean permanentes.





8- Al ejecutar `phackeame_patched.exe`, cualquier usuario y contraseña activa el mensaje de éxito: "¡Lo lograste!".