Writeup: Análisis de archivo.exe

Introducción

Este writeup documenta el análisis de simpleServer.exe, un ejecutable sospechoso analizado mediante ingeniería inversa. El objetivo era extraer información clave como:

1-Dominio malicioso
2-Credenciales incrustadas (usuario y contraseña)
3-Método de cifrado utilizado

Para ello, se emplearon diversas herramientas de análisis estático y dinámico, destacando Radare2 y Python para la extracción y descifrado de datos.

Herramientas Utilizadas

Herramienta y comandos	Función
Radare2 (r2)	Ingeniería inversa del ejecutable, extracción de cadenas, desensamblado y análisis de funciones.
Strings (strings)	Extracción de texto legible del ejecutable para identificar información clave.
Python	Creación de un script para descifrar cadenas cifradas con XOR.
Grep (grep)	Filtrado de cadenas relevantes como dominios, contraseñas y comandos sospechosos.

1. Extracción de Cadenas de Texto

¿Por qué extraer cadenas?

Las cadenas de texto en un ejecutable pueden revelar información sensible, como:

Comandos internos (net user, cmd /c)

Dominios maliciosos

Credenciales embebidas

Mensajes de error que indican funcionalidad

Comando Utilizado

strings simpleServer.exe > extracted_strings.txt

Esto genera un archivo extracted_strings.txt con todas las cadenas legibles.

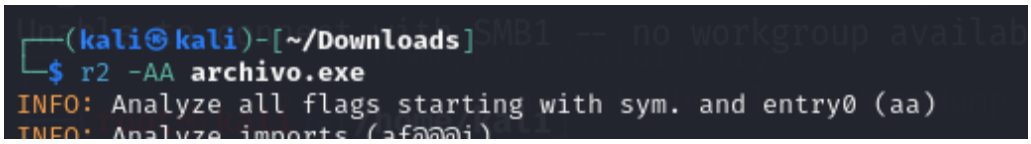
Filtrar información clave con grep

Comando Utilizado

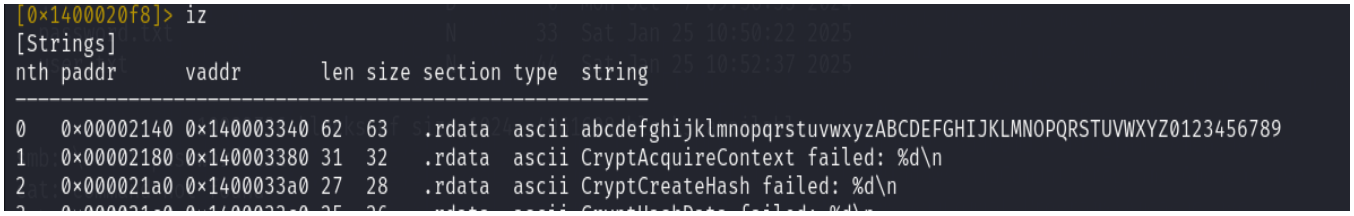
grep -Ei 'user|pass|auth|domain|info.txt' extracted_strings.txt

2. Análisis del Código en Radare2

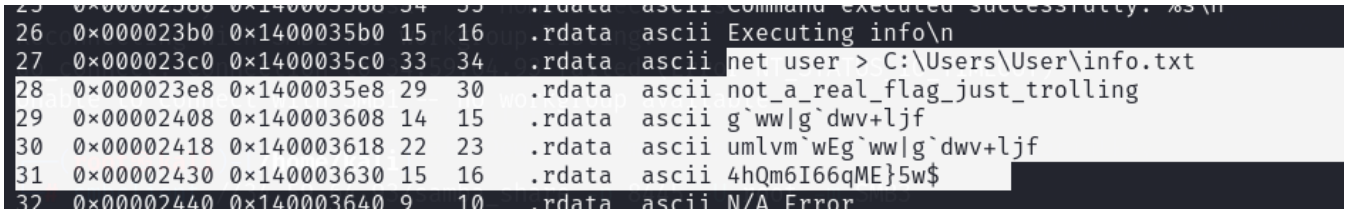
r2 -AA archivo.exe



Esto muestra todas las cadenas encontradas en el binario, confirmando la presencia de



Resultado relevante:

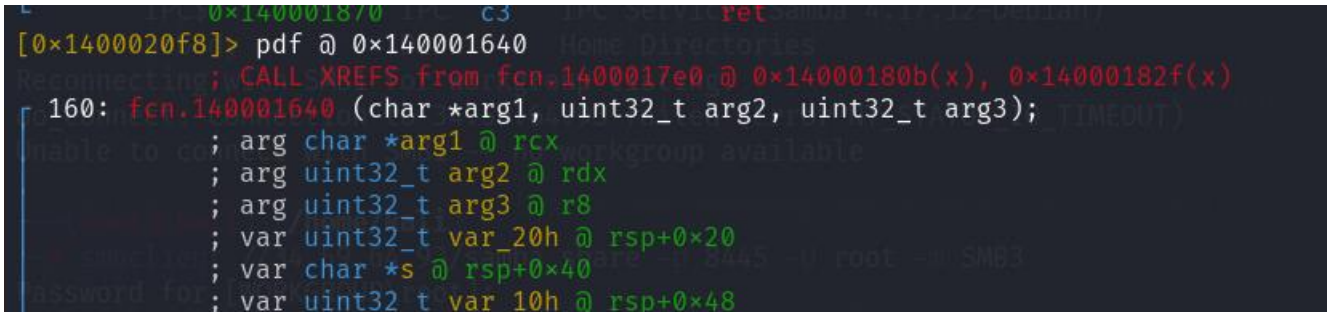


Esto sugiere que el binario obtiene información de usuarios y contiene posibles cadenas cifradas.

3. Análisis de la Función de Descifrado

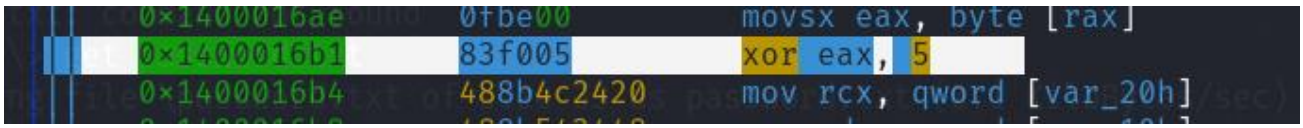
Se inspecciona la función identificada con:

pdf @ 0x140001640



Hallazgos Clave en la Función

- Llama a strlen() para calcular la longitud de la cadena.
- Recorre la cadena carácter por carácter.
- Aplica una operación XOR con la constante 0x05:

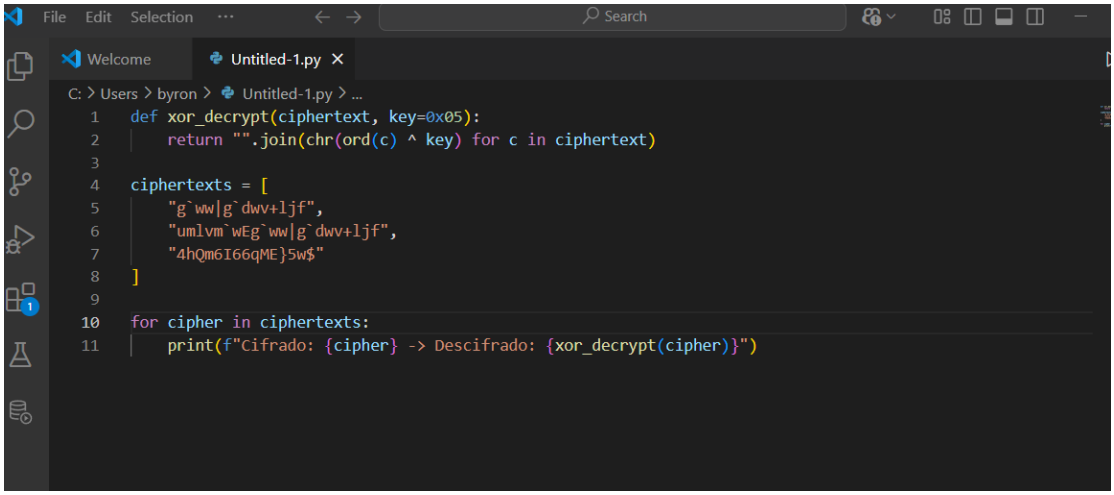


Esto confirma que las cadenas están cifradas con XOR y la clave 0x05.

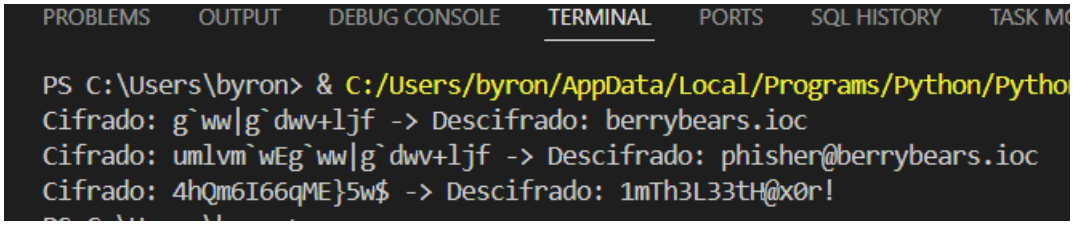
4. Descifrado de las Cadenas en Python

Con la clave XOR identificada, se creó un script en Python para recuperar los valores en texto plano.

Código Python para Descifrar



Resultados Obtenidos



5. Conclusión Final

Pregunta	Respuesta
¿Cuál es el dominio malicioso?	**berrybears.ioc**
¿Cuál es el usuario presente?	**phisher@berrybears.ioc**
¿Cuál es la contraseña?	**1mTh3L33tH@x0r!**

Resumen del Método Utilizado

1. **Extracción de cadenas (strings)** → Identificación de texto cifrado.
2. **Análisis en Radare2** → Ubicación de la función que descifra las cadenas.
3. **Desensamblado de código (pdf @ dirección)** → Identificación del uso de XOR con 0x05.

4. **Descifrado en Python** → Recuperación de los valores en texto claro.

¿Qué significa este hallazgo?

- **El malware se comunica con berrybears.ioc**, indicando un servidor de comando y control (C2).
- **Las credenciales embebidas (phisher@berrybears.ioc) pueden ser usadas para autenticación.**
- **La contraseña 1mTh3L33tH@x0r! probablemente se usa para acceder al servidor o cifrar datos antes de enviarlos.**

Conclusión Final

El análisis de archivo.exe demostró el uso de **XOR para cifrado de datos** y la existencia de un servidor remoto que recibe información potencialmente robada. Gracias a herramientas como **Radare2, Strings, Grep y Python**, se logró descifrar el contenido del malware y obtener respuestas clave.Este enfoque puede aplicarse a otros binarios maliciosos para extraer credenciales, identificar servidores de ataque y comprender mejor la funcionalidad del malware.