

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO



**FACULTAD DE
CIENCIAS**



**CIENCIAS DE LA
COMPUTACIÓN**

CRIPTOGRAFÍA Y SEGURIDAD

PRACTICA 01: ROT13 EN USSEERASSIST

**ALUMNO: GARCÍA HERNÁNDEZ JOSÉ
ANTONIO**

**-PAULO SANTIAGO DE JESÚS
CONTRERAS FLORES
-SÁNCHEZ NERI JESÚS TONATIHU
-GALINDO PEREZ IVAN DANIEL**

SEMESTRE 2025-2

CRIPTOGRAFÍA Y SEGURIDAD
CONTRERAS FLORES PAULO SANTIAGO DE JESÚS
JESÚS TONATIHU SÁNCHEZ NERI

Práctica 01:

ROT13 en UserAssist

García Hernández José Antonio

Introducción

El presente reporte detalla el proceso de análisis forense realizado sobre claves del registro de Windows, específicamente las entradas de la clave UserAssist. El objetivo principal fue identificar posibles programas ejecutados por un usuario, prestando especial atención a la detección de software potencialmente malicioso. La clave UserAssist almacena información sobre los programas ejecutados con mayor frecuencia y los accesos directos utilizados, lo que la convierte en una fuente valiosa para la investigación forense. En esta práctica, se aplicó el algoritmo de cifrado ROT13 para descifrar los nombres de los programas almacenados en las entradas UserAssist.

Desarrollo

Debido a inconvenientes técnicos con la máquina virtual de Windows 10 proporcionada en Classroom para esta práctica, el análisis se realizó directamente sobre un conjunto de entradas UserAssist previamente extraídas. El proceso de descifrado se llevó a cabo utilizando dos métodos diferentes, ambos implementados en un entorno Linux:

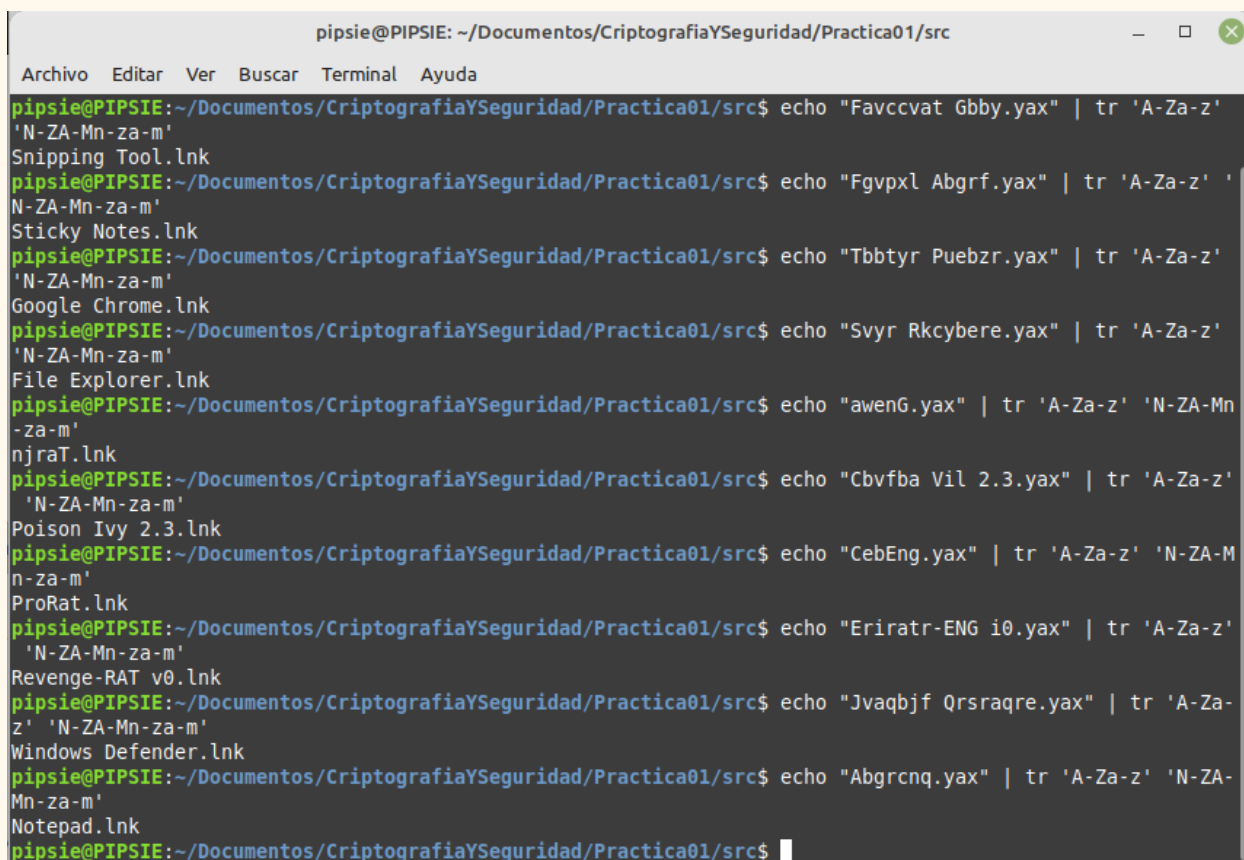
1. Descifrado con tr (Transliterate):

El comando tr es una utilidad estándar de Linux que nos permite realizar sustituciones de caracteres. En este caso, se utilizó para implementar el cifrado ROT13 mediante la sustitución de letras del alfabeto por aquellas que se encuentran 13 posiciones adelante. Debido a la naturaleza del algoritmo ROT13, la misma función de cifrado puede utilizarse para descifrar.

El comando utilizado fue:

```
$ echo "TextoEncriptado" | tr 'A-Za-z' 'N-ZA-Mn-za-m'
```

El comando se aplicó individualmente a cada una de las entradas del UserAssist, reemplazando “TextoEncriptado” con los valores codificados. Cada comando se ejecutó directamente en la terminal, y el resultado descifrado se registró para su posterior análisis.



```

pipsie@PIPSIE: ~/Documentos/CriptografiaYSeguridad/Practica01/src
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
pipsie@PIPSIE:~/Documentos/CriptografiaYSeguridad/Practica01/src$ echo "Favccvat Gbby.yax" | tr 'A-Za-z' 'N-ZA-Mn-za-m'
Snipping Tool.lnk
pipsie@PIPSIE:~/Documentos/CriptografiaYSeguridad/Practica01/src$ echo "Fgvpxl Abgrf.yax" | tr 'A-Za-z' 'N-ZA-Mn-za-m'
Sticky Notes.lnk
pipsie@PIPSIE:~/Documentos/CriptografiaYSeguridad/Practica01/src$ echo "Tbbyr Puebzr.yax" | tr 'A-Za-z' 'N-ZA-Mn-za-m'
Google Chrome.lnk
pipsie@PIPSIE:~/Documentos/CriptografiaYSeguridad/Practica01/src$ echo "Svyr Rkcybere.yax" | tr 'A-Za-z' 'N-ZA-Mn-za-m'
File Explorer.lnk
pipsie@PIPSIE:~/Documentos/CriptografiaYSeguridad/Practica01/src$ echo "awenG.yax" | tr 'A-Za-z' 'N-ZA-Mn-za-m'
njraT.lnk
pipsie@PIPSIE:~/Documentos/CriptografiaYSeguridad/Practica01/src$ echo "Cbvfba Vil 2.3.yax" | tr 'A-Za-z' 'N-ZA-Mn-za-m'
Poison Ivy 2.3.lnk
pipsie@PIPSIE:~/Documentos/CriptografiaYSeguridad/Practica01/src$ echo "CebEng.yax" | tr 'A-Za-z' 'N-ZA-Mn-za-m'
ProRat.lnk
pipsie@PIPSIE:~/Documentos/CriptografiaYSeguridad/Practica01/src$ echo "Eriratr-ENG i0.yax" | tr 'A-Za-z' 'N-ZA-Mn-za-m'
Revenge-RAT v0.lnk
pipsie@PIPSIE:~/Documentos/CriptografiaYSeguridad/Practica01/src$ echo "Jvaqbjf Qrsraqre.yax" | tr 'A-Za-z' 'N-ZA-Mn-za-m'
Windows Defender.lnk
pipsie@PIPSIE:~/Documentos/CriptografiaYSeguridad/Practica01/src$ echo "Abgrcnq.yax" | tr 'A-Za-z' 'N-ZA-Mn-za-m'
Notepad.lnk
pipsie@PIPSIE:~/Documentos/CriptografiaYSeguridad/Practica01/src$

```

2. Descifrado con Script de Python:

Para automatizar el proceso y asegurar la consistencia de los resultados, se desarrolló un script de Python que implementa la función ROT13.

```
def rot13(text):
    result = ''
    for char in text:
        if 'a' <= char <= 'z':
            result += chr((ord(char) - ord('a') + 13) % 26 + ord('a'))
        elif 'A' <= char <= 'Z':
            result += chr((ord(char) - ord('A') + 13) % 26 + ord('A'))
        else:
            result += char
    return result

# Lista de nuestros textos cifrados
texts = ["Cnvag.yax", "Favccvat Gbby.yax", "Fgvpxl Abgrf.yax", "Tbbyr Puebzt.yax", "Svyr Rkcybere.yax",
"awenG.yax", "Cvbfba Vil 2.3.yax", "CebEng.yax", "Eratr-ENG i0.yax", "Jvaqbjf Qrsraqre.yax", "Abgrcnq.yax"]

decoded_texts = [rot13(text) for text in texts]

for original, decoded in zip(texts, decoded_texts):
    print(f"Cifrado: {original} -> Descifrado: {decoded}")
```

El script define una función rot13 que realiza el cifrado/descifrado ROT13. Posteriormente, se crea una lista con las cadenas a descifrar, se aplica la función a cada elemento de la lista y se imprime el resultado original y descifrado para su fácil visualización.

```
pipsie@PIPSIE:~/Documentos/CriptografiaYSeguridad/Practica01/src$ python3 rot13.py
Cifrado: Cnvag.yax -> Descifrado: Paint.lnk
Cifrado: Favccvat Gbby.yax -> Descifrado: Snipping Tool.lnk
Cifrado: Fgvpxl Abgrf.yax -> Descifrado: Sticky Notes.lnk
Cifrado: Tbbyr Puebzt.yax -> Descifrado: Google Chrome.lnk
Cifrado: Svyr Rkcybere.yax -> Descifrado: File Explorer.lnk
Cifrado: awenG.yax -> Descifrado: njraT.lnk
Cifrado: Cbvfba Vil 2.3.yax -> Descifrado: Poison Ivy 2.3.lnk
Cifrado: CebEng.yax -> Descifrado: ProRat.lnk
Cifrado: Eratr-ENG i0.yax -> Descifrado: Revenge-RAT v0.lnk
Cifrado: Jvaqbjf Qrsraqre.yax -> Descifrado: Windows Defender.lnk
Cifrado: Abgrcnq.yax -> Descifrado: Notepad.lnk
pipsie@PIPSIE:~/Documentos/CriptografiaYSeguridad/Practica01/src$
```

Ambos métodos de descifrado produjeron los mismos resultados, validando la integridad del proceso.

3. Análisis de Resultados Descifrados

Tras el descifrado, se procedió a analizar los nombres de los programas revelados. El objetivo era identificar aquellos que pudieran ser indicativos de actividad maliciosa. Los programas identificados fueron los siguientes:

- Paint.lnk
- Snipping Tool.lnk
- Sticky Notes.lnk
- Google Chrome.lnk
- File Explorer.lnk
- njraT.lnk
- Poison Ivy 2.3.lnk
- ProRat.lnk
- Revenge-RAT v0.lnk

Se detectaron entradas sospechosas:

- Poison Ivy 2.3.lnk: Troyano de acceso remoto (RAT) conocido por su capacidad de ocultarse y persistir en el sistema, permitiendo a un atacante controlar el equipo de forma remota.
- ProRat.lnk: Otro troyano de acceso remoto (RAT), que permite el control remoto del sistema, administración de archivos, control del escritorio, robo de contraseñas y otras funciones maliciosas.
- Revenge-RAT v0.lnk: Troyano de acceso remoto (RAT) que otorga a un atacante control remoto del equipo infectado, permitiendo el acceso al sistema de archivos, la ejecución de comandos y la captura de pantalla.

La presencia de estos RATs es un fuerte indicio de actividad maliciosa en el sistema. La entrada njraT.lnk también debería investigarse para determinar si se trata de un programa legítimo o de una herramienta maliciosa disfrazada.

Cuestionario

1. ¿Encontró indicios de la ejecución de software malicioso?

a. En caso afirmativo, liste el software considerado malicioso y explique brevemente para que se usa cada uno.

Sí, encontramos indicios de la posible ejecución de software malicioso. Específicamente, se identificó la entrada Revving-RAT v0.yax.

- **Revenge-RAT:** "RAT" significa "Remote Access Trojan" (Troyano de Acceso Remoto). Los RATs son herramientas maliciosas que permiten a un atacante controlar remotamente un equipo infectado. Sus funciones pueden ser:

- Acceso al sistema de archivos
- Ejecución de comandos
- Captura de pantalla y grabación de video
- Robo de contraseñas
- Instalación de otros programas maliciosos
- **ProRat:** Otro troyano de acceso remoto (RAT). Similar a Revving-RAT, ProRat permite el control remoto de un sistema infectado. Algunas de sus características incluyen:
 - Administración de archivos.
 - Control del escritorio.
 - Robo de contraseñas.
 - Ataques DDoS (Denegación de Servicio Distribuido).
 - Keylogging (Registro de pulsaciones del teclado).
- **Poison Ivy 2.3:** Otro troyano de acceso remoto (RAT) ampliamente utilizado. Poison Ivy es conocido por su capacidad de ocultarse y persistir en el sistema. Algunas de sus características incluyen:
 - Acceso y control del sistema de archivos.
 - Captura de credenciales.
 - Grabación de audio y video.
 - Redirección de tráfico de red.

2. ¿En qué directorios se encontraba cada link (.lnk) de los programas de reciente uso listados en el UserAssist? ¿Por qué cree que se encontraban en esas ubicaciones?

Los links (.lnk) de los programas de reciente uso listados en el UserAssist se encontraban en los siguientes directorios:

- Accesorios (Accessories)
- Barra de tareas (TaskBar)
- Directorio raíz de algunos GUIDs (Identificadores Únicos Globales)

¿Por qué se encontraban allí?

- Accesorios: Los accesos directos a programas en la carpeta "Accesorios" están allí porque son parte de las herramientas básicas que Windows proporciona y los usuarios suelen acceder a ellos desde el menú Inicio.
- Barra de tareas: Los accesos directos en la barra de tareas están ahí porque el usuario ha elegido anclar estos programas para tener un acceso rápido y fácil a ellos.

- Directorio raíz de algunos GUIDs: La presencia de accesos directos en directorios con nombres de GUIDs nos puede indicar que el programa fue instalado por un instalador específico, y el GUID representa un identificador único para la aplicación o un componente del sistema. También, puede indicar que el programa ha creado su propia carpeta dentro de un directorio del sistema, para organizar sus propios archivos y recursos.

3. ¿Por qué cree que en un análisis previo no se encontraron los programas instalados en el equipo asegurado?, y ¿Por qué cree que, aunque se tienen los rastros del acceso al software malicioso, ya no se encuentra ese software instalado en el equipo?

Pueden ser varias las razones posibles por las que un análisis previo no detectó los programas instalados:

- El software fue desinstalado antes del análisis: El usuario pudo haber desinstalado el software después de usarlo, pero antes de que se realizara la adquisición forense del sistema.
- El software es portable y no requiere instalación: Algunos programas se ejecutan directamente sin necesidad de instalación, por lo que no dejan rastros en la lista de programas instalados. Sin embargo, aún pueden dejar rastros de ejecución en UserAssist.
- El software está oculto o disfrazado: El malware intenta ocultarse de los análisis antivirus y de la lista de programas instalados.

¿Por qué hay rastros en UserAssist pero el software ya no está?

- Desinstalación: Como mencionamos anteriormente, el software pudo haber sido desinstalado. UserAssist conserva los rastros de ejecución incluso después de la desinstalación del programa, hasta que las entradas sean eliminadas o sobrescritas.
- Eliminación por software de seguridad: Un programa antivirus o antimalware pudo haber detectado y eliminado el software malicioso, pero UserAssist aún conserva los registros de su ejecución anterior.
- El software era un ejecutable temporal: El software pudo haber sido ejecutado directamente desde un medio extraíble (como una USB) o descargado temporalmente y luego eliminado.

4. ¿Qué relación tiene esta práctica con el análisis forense?

Esta práctica está directamente relacionada con el análisis forense digital, puesto que se centra en:

- **Identificación de actividad del usuario:** UserAssist nos proporciona información valiosa sobre qué programas ha estado ejecutando el usuario, cuándo los ha ejecutado y con qué frecuencia.
- **Detección de software malicioso:** Con el análisis de las entradas UserAssist se puede revelar la ejecución de programas sospechosos que podrían indicar una infección de malware.
- **Reconstrucción de eventos:** La información en UserAssist nos puede ayudar a reconstruir la secuencia de eventos que ocurrieron en el sistema, lo que es fundamental en una investigación forense.
- **Obtención de evidencias:** Las entradas UserAssist pueden ser utilizadas como evidencia en un caso legal o investigación interna.

5. Investigue por qué Windows usa ROT13 para cifrar esta información e indíquelo en su reporte

Windows utiliza ROT13 para "cifrar" la información en la clave UserAssist no por razones de seguridad robusta, sino más bien como una forma sencilla de ofuscación. Algunas razones son:

- Impedir la lectura casual: ROT13 evita que usuarios casuales (o scripts simples) lean directamente la información en UserAssist y comprendan fácilmente qué programas se están ejecutando.
- No es una medida de seguridad real: ROT13 es un cifrado muy débil y fácil de revertir. No está diseñado para proteger la información contra un atacante determinado.
- Compatibilidad y rendimiento: ROT13 es un algoritmo muy simple de implementar y rápido de ejecutar, lo que minimiza el impacto en el rendimiento del sistema.
- Propósito principal: Usabilidad, no seguridad: UserAssist está diseñado principalmente para mejorar la experiencia del usuario, no para proteger información sensible. La ofuscación ROT13 es suficiente para ocultar la información de miradas indiscretas, pero no está diseñada para resistir un análisis forense.

6. Investigue sobre las siguientes herramientas UserAssistView v1.02 y UserAssist v2.6.0, y emita un comentario sobre su utilidad.

UserAssistView (NirSoft): Es una herramienta gratuita desarrollada por NirSoft que permite visualizar y analizar fácilmente la información almacenada en la clave UserAssist del registro de Windows.

- UserAssistView v1.02: Una versión más antigua de la herramienta.
- UserAssistView v2.6.0: Una versión más reciente con mejoras y características adicionales.

Ambas versiones son muy útiles porque simplifican la visualización, pues presentan la información de UserAssist de una manera organizada y fácil de entender, lo cual nos ahorra tiempo y esfuerzo, de igual forma cuentan con un filtro para encontrar info en grandes conjuntos de datos. También nos permiten descifrar automáticamente las cadenas ROT13, lo que facilita mucho la identificación de los programas ejecutados. Asimismo nos muestran información como: el número de ejecuciones de un programa y la última fecha de ejecución, lo que puede ayudar a determinar la importancia de un programa en la actividad del usuario.

Conclusión

El análisis de las claves UserAssist del registro de Windows demostró ser una técnica valiosa para identificar programas ejecutados en un sistema, permitiéndonos detectar software malicioso. El uso del descifrado ROT13 permitió revelar los nombres de los programas, lo que a su vez llevó a la identificación de múltiples amenazas de seguridad, como la presencia de los troyanos de acceso remoto Poison Ivy 2.3, ProRat y Revenge-RAT. La identificación de estos programas maliciosos nos sugiere que el sistema comprometido puede estar bajo el control de un atacante remoto.

En base a los resultados obtenidos, podemos concluir que es crucial realizar una investigación más exhaustiva para determinar el alcance de la infección, identificar las acciones realizadas por el atacante y tomar las medidas de remediación necesarias. Esto puede incluir el análisis de otros artefactos forenses, la búsqueda de archivos maliciosos adicionales, la revisión de los registros de red y la implementación de medidas de seguridad para prevenir futuras infecciones. La información proporcionada por las claves UserAssist, junto con otras técnicas de análisis forense, puede ayudar a reconstruir la secuencia de eventos que ocurrieron en el sistema y a comprender mejor las acciones realizadas por el atacante.

Referencias

- Carvey H. (2014). *Windows Forensic Analysis Toolkit*. Syngress.
- Hale M., Case A., Levy J. & Walters A. (2014). *The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory*. Wiley.