Asignatura	Datos del alumno	Fecha
Seguridad en los Sistemas de Información	Apellidos: Jimenez Acosta	
	Nombre: Ronaldo	

Actividad 2. Vectores de ataque

El objetivo principal de esta actividad es analizar un vector de ataque específico, en este caso, el uso de software pirata, documentando cómo este tipo de prácticas pueden facilitar la explotación de vulnerabilidades en sistemas y comprometer la seguridad de los usuarios. A través de este trabajo, se busca detallar cómo se puede estructurar un ciberdelito utilizando código, evaluando su funcionamiento y sus riesgos asociados.

Un vector de ataque es el camino que sigue un ciberdelincuente para acceder al activo objetivo, explotando las vulnerabilidades existentes. Estos vectores pueden variar ampliamente dependiendo del tipo de ataque y el objetivo. Entre los ciberdelitos más comunes encontramos **phishing, smishing, inserción de malware, keyloggers, botnets, pharming, carding, skimming, fraude del CEO, y más.**

En todos estos casos, el vector de ataque juega un papel clave, ya que es el primer paso que permite a los delincuentes ingresar al sistema objetivo. Por ejemplo, en ataques de ransomware, el vector de ataque suele ser la **ingeniería social**, convenciendo a un usuario desprevenido de hacer clic en un enlace malicioso o descargar un archivo, que luego ejecuta el malware en el sistema.

Este trabajo explora un vector de ataque particular basado en **el uso de software pirata**, donde el cibercriminal ofrece una supuesta activación gratuita de una aplicación a través de un archivo ejecutable que, en segundo plano, ejecuta acciones dañinas.

Asignatura	Datos del alumno	Fecha
Seguridad en los Sistemas de Información	Apellidos: Jimenez Acosta	
	Nombre: Ronaldo	

Análisis del Vector de Ataque: Uso de Software Pirata

En este trabajo se simula un ataque sencillo a través de un programa en Python que se presenta como un "activador gratuito de software". El código simula el proceso de activación de una aplicación, pero de manera oculta crea y ejecuta un archivo .bat que abre una página web sin el consentimiento del usuario.

Funcionamiento del código.

El código realiza las siguientes acciones:

- **Interfaz gráfica (GUI):** El programa utiliza el módulo tkinter para crear una interfaz sencilla con un botón que ofrece la activación gratuita de una aplicación.
- **Creación de un archivo .bat:** Cuando el usuario hace clic en el botón de "Activar", el programa crea un archivo .bat en el directorio de documentos del usuario.
- **Ejecución del .bat:** El archivo .bat ejecuta un comando para abrir automáticamente una pestaña en el navegador web (en este caso, Google Chrome) apuntando a una URL especificada.
- **Simulación de activación:** El usuario recibe un mensaje indicando que la activación fue "exitosa", mientras que en segundo plano el sistema ha sido manipulado.

Este tipo de ataques refleja un vector de ataque basado en **ingeniería social**, donde se engaña al usuario para que ejecute un programa que parece inofensivo, pero que en realidad compromete la seguridad del sistema.

Posibles riesgos:

- **Inserción de malware:** Aunque en este ejemplo solo se abre una página web, el archivo .bat podría ser modificado para descargar y ejecutar malware en el sistema, como troyanos, keyloggers, ransomware, etc.

- **Robo de información personal:** El atacante podría redirigir al usuario a páginas web maliciosas diseñadas para robar información personal o credenciales de acceso.
- **Control remoto:** A través de la ejecución de scripts más avanzados, se podría obtener control remoto del sistema, convirtiendo al equipo en parte de una botnet.
- **Modificación del sistema:** Este código demuestra cómo un software aparentemente legítimo puede realizar cambios en el sistema sin el conocimiento del usuario.

¿Es fácil realizar un ciberdelito con esta información?

Sí, con acceso a información básica de programación y herramientas como Python, es relativamente sencillo implementar scripts maliciosos que exploten la falta de conocimiento o la confianza del usuario. Herramientas simples como la creación de archivos .bat pueden usarse para desencadenar una cadena de eventos que comprometan la seguridad de un sistema.

Prevención

Para evitar este tipo de ataques, es importante:

- **Educación al usuario:** Los usuarios deben ser conscientes de los riesgos de usar software pirata. El software legítimo generalmente viene con soporte, actualizaciones de seguridad y garantías de integridad.
- **Software legítimo y actualizado:** Usar software oficial y mantenerlo actualizado reduce las posibilidades de explotación de vulnerabilidades conocidas.
- **Antivirus y firewall:** Tener protección antimalware activa y un firewall configurado correctamente ayuda a detectar actividades sospechosas.

- ✧ **No descargar archivos de fuentes no confiables:** Los usuarios deben evitar descargar activadores, cracks o cualquier archivo de fuentes no verificadas, ya que a menudo están empaquetados con malware.
- ✧ **Ejecutar auditorías de seguridad:** Las organizaciones deben implementar auditorías regulares para detectar software no autorizado en sus redes.

Nota: Para complementar este taller, he creado un video explicativo donde detallo el funcionamiento del programa y los riesgos asociados al uso de software pirata. El video está subido a YouTube como público pero oculto, ya que, debido a las limitaciones de los recursos de mi equipo, la grabación tuvo algunos problemas de rendimiento. **Aunque el audio se escucha correctamente, la imagen puede quedarse pegada en algunos momentos.** A pesar de estos inconvenientes, el contenido del video cubre de manera clara el análisis presentado aquí.

Puedes acceder al video en el siguiente enlace:
https://youtu.be/ssiOI7KxQSE
Además, todo el código esta disponible en el siguiente repositorio de GitHub:
https://github.com/JimcostDev/Python_Ejercicios/blob/master/ejercicios/attack_vectors/main.py