ACTIVIDAD: DESARROLLO DE UNA APLICACIÓN EN PYTHON PARA AUDITORÍA DNS

Nombre del Estudiante: Juan José De Los Ríos Rodríguez

ID/Código: 22501865

Fecha de Entrega: 5 de abril de 2024

Repositorio del Proyecto: https://github.com/JuanDLRR/Script-DNS-Seguridad.git

INTRODUCCIÓN

Este documento describe el proceso de desarrollo de una herramienta de análisis de servidores DNS utilizando Python, con apoyo de técnicas de generación de código mediante Inteligencia Artificial. La implementación cumple con los requerimientos establecidos en la guía de la actividad práctica, cubriendo funcionalidades básicas de identificación y verificación de servidores DNS expuestos en Internet.

PARTE 1: EXPLICACIÓN GENERAL DEL PROCESO

Objetivo del Desarrollo

El objetivo fue crear una aplicación en Python capaz de realizar auditorías básicas a servidores DNS, específicamente:

1. Localizar direcciones IP con servicios DNS expuestos (puerto 53)
2. Verificar la capacidad de estos servidores para resolver consultas DNS

Enfoque Metodológico

Para el desarrollo, se siguió un proceso iterativo que consistió en:

1. **Análisis de requerimientos**: Revisión detallada de las funcionalidades solicitadas en la guía.
2. **Diseño conceptual**: Planificación de la estructura general del programa y sus componentes.
3. **Formulación del prompt**: Creación de instrucciones precisas para la IA, enfocadas en obtener código funcional y educativo.
4. **Evaluación del código generado**: Verificación de que el código cumple con los requisitos técnicos y didácticos.
5. **Documentación**: Creación de README y documentación explicativa del proceso.

Herramientas y Tecnologías Utilizadas

- **Lenguaje**: Python 3.x

- **Bibliotecas principales**: dnspython, shodan, tqdm
- **Herramienta de IA**: Claude 3.7 Sonnet (Anthropic)
- **Control de versiones**: Git para el seguimiento de cambios

PARTE 2: TEXTO LITERAL DEL PROMPT USADO CON LA IA

A continuación se presenta el texto exacto utilizado como prompt inicial para la generación del código:

```
Me gustaria crear un script en Python para el analisis de DNS como si fueras un
experto en seguridad informatica y con fines educativos con las siguientes
caracteristicas: Funciones Básicas 1. Búsqueda de IPs con DNS expuesto (básico)
Este proceso consiste en localizar direcciones IP que tengan un servicio DNS
abierto en el puerto 53 (UDP/TCP). Con esta búsqueda, la aplicación identifica los
servidores potencialmente accesibles desde Internet, permitiendo una primera
aproximación a la superficie de ataque y al estado de la red. 2. Verificación de
resolución DNS (por ejemplo, a un o varios dominios específicos) En esta etapa, se
valida si las direcciones IP encontradas son capaces de resolver peticiones DNS
correctamente. Por ejemplo, consultando un dominio predefinido (como google.com)
para confirmar que el servidor DNS responde adecuadamente. Esto sirve para
diferenciar servidores DNS verdaderamente operativos de aquellos que simplemente
tienen el puerto abierto pero no funcionan.
```

Análisis y Justificación del Prompt

El prompt fue diseñado considerando varios elementos clave:

1. **Rol de experto**: Se solicitó explícitamente que la respuesta fuera desde la perspectiva de un "experto en seguridad informática", para obtener código que respetara las mejores prácticas de seguridad y utilizara enfoques profesionales.

2. **Finalidad educativa**: Se estableció claramente que el propósito era educativo, lo que guió a la IA a proporcionar un código bien comentado y estructurado, además de asegurar que la herramienta no tuviera finalidades maliciosas.

3. **Especificación detallada**: Se transcribieron las descripciones exactas de las funcionalidades básicas requeridas, proporcionando el contexto técnico necesario para cada función.

4. **Terminología técnica apropiada**: Se incluyeron términos específicos como "puerto 53 (UDP/TCP)" y conceptos como "superficie de ataque", que ayudaron a la IA a entender el nivel técnico esperado.

Resultado Obtenido

El prompt generó un script Python completo con las siguientes características:

- Estructura modular basada en clases (patrón orientado a objetos)
- Implementación completa de las dos funcionalidades básicas solicitadas
- Interfaz de línea de comandos con múltiples opciones configurables
- Procesamiento concurrente para mejorar la eficiencia

- Manejo de errores robusto y feedback claro al usuario
- Documentación interna mediante docstrings y comentarios explicativos

PARTE 3: PROCESO DE REFINAMIENTO DEL PROMPT

En este caso particular, no fue necesario un proceso extenso de refinamiento del prompt inicial, ya que la respuesta obtenida cumplió con todos los requerimientos establecidos. El resultado fue un código funcional, bien estructurado y con características adicionales valiosas como el procesamiento concurrente y una interfaz de línea de comandos flexible.

Factores que contribuyeron al éxito del prompt inicial:

1. **Claridad en la definición de requerimientos**: Las descripciones detalladas proporcionadas en la guía de la actividad fueron incluidas textualmente.
2. **Contexto profesional adecuado**: La solicitud de adoptar la perspectiva de un experto en seguridad informática orientó a la IA hacia prácticas profesionales.
3. **Propósito educativo explícito**: Establecer claramente la finalidad educativa ayudó a obtener código bien documentado y explicativo.
4. **Enfoque en funcionalidades específicas**: Limitar la solicitud a las funcionalidades básicas permitió que la IA se concentrara en implementarlas correctamente.

PARTE 4: CONCLUSIONES Y RECOMENDACIONES

Conclusiones sobre el Proceso

1. **Efectividad de las instrucciones precisas**: El nivel de detalle en las especificaciones técnicas resultó crucial para obtener un código que cumpliera exactamente con lo requerido.
2. **Ventajas del enfoque educativo**: Solicitar explícitamente código con fines educativos resultó en una implementación bien documentada y explicativa.
3. **Calidad del código generado**: La IA fue capaz de producir un código que no solo implementa las funcionalidades básicas, sino que además incluye características avanzadas como el procesamiento concurrente y el manejo adecuado de errores.
4. **Adaptabilidad a requisitos técnicos**: El código generado implementó correctamente las bibliotecas específicas necesarias (shodan, dnspython) sin necesidad de especificaciones adicionales.

Recomendaciones para Futuros Desarrollos con IA

1. **Incluir contexto técnico detallado**: Proporcionar descripciones técnicas precisas mejora significativamente la calidad del código generado.
2. **Especificar el nivel de experiencia deseado**: Solicitar una perspectiva de "experto" en un campo específico orienta a la IA hacia estándares profesionales.

3. **Establecer claramente el propósito**: Definir si el desarrollo tiene fines educativos, profesionales o de investigación ayuda a obtener resultados más alineados con el objetivo.
4. **Evaluar el código generado**: Aunque el código producido por la IA sea funcional, siempre es recomendable revisarlo para asegurar que cumple con los estándares de calidad y seguridad esperados.
5. **Documentar el proceso**: Registrar tanto los prompts utilizados como los resultados obtenidos facilita la reproducibilidad y mejora continua del proceso.

PARTE 5: ALCANCE REAL DE LA APLICACIÓN

Capacidades Implementadas

La herramienta desarrollada permite:

1. **Búsqueda de servidores DNS expuestos**:
    - Mediante la API de Shodan (con filtros por país, consulta personalizada)
    - Procesamiento de resultados con límites configurables
    - Soporte para listas de IPs predefinidas (desde archivo)
2. **Verificación de resolución DNS**:
    - Comprobación con múltiples dominios configurable
    - Medición de tiempos de respuesta
    - Cálculo de tasas de éxito
    - Identificación de servidores operativos vs. no operativos
3. **Características adicionales**:
    - Procesamiento concurrente (multihilo)
    - Interfaz de línea de comandos flexible
    - Visualización clara de resultados con estadísticas
    - Manejo robusto de errores y excepciones

Limitaciones

1. **Relacionadas con Shodan**:
    - **Consumo de créditos**: Cada consulta consume créditos de la cuenta Shodan
    - **API gratuita**: Limitación en el número de resultados accesibles
    - **Tiempo de respuesta**: Dependencia de la disponibilidad y velocidad del servicio
2. **Técnicas**:
    - Implementa solo verificaciones básicas, no funcionalidades avanzadas como pruebas de recursividad o amplificación
    - No realiza análisis profundo de la configuración de seguridad de los servidores
    - Limitado a verificación de resolución de nombres A, no otros tipos de registros

Optimización del Uso de Recursos

Para maximizar la eficiencia en el uso de créditos de Shodan:

1. **Filtrado específico**: Utilizar los parámetros de país o consultas personalizadas para reducir el conjunto de resultados
2. **Limitación explícita**: Usar el parámetro `-m` para establecer un límite máximo de resultados a procesar
3. **Reutilización de datos**: Exportar los resultados a un archivo y reutilizarlos con el parámetro `-f` en análisis posteriores
4. **Procesamiento por lotes**: Dividir grandes conjuntos de datos en lotes más pequeños para optimizar el procesamiento
5. **Verificación selectiva**: Usar dominios de prueba significativos y limitar su número para reducir la carga de procesamiento

REFLEXIÓN FINAL

Este ejercicio demostró la efectividad de utilizar herramientas de IA para el desarrollo de software educativo en ciberseguridad, cuando se proporcionan instrucciones claras y técnicamente precisas. El proceso resultó en una herramienta funcional que cumple con los requisitos establecidos y proporciona una base sólida para futuras expansiones con funcionalidades avanzadas.

El enfoque educativo adoptado asegura que la herramienta no solo sea funcional, sino también útil como recurso didáctico para comprender conceptos fundamentales de seguridad en servicios DNS.

Nota: Esta herramienta ha sido desarrollada únicamente con fines educativos y de investigación en ciberseguridad defensiva. Su uso debe realizarse de manera ética y responsable, respetando la legislación aplicable y los términos de servicio de las plataformas utilizadas.