



Ejercicio Práctico (Nivel Medio): Analizador de Cabeceras HTTP

Descripción

En este ejercicio crearás una herramienta en Python capaz de enviar una petición HTTP a un sitio web y analizar las cabeceras (headers) de la respuesta. Este tipo de herramienta es muy utilizada en fases de reconocimiento en pruebas de penetración (recon), ya que permite obtener información del servidor, como tecnologías usadas, políticas de seguridad, cookies, y más.

Objetivo del ejercicio

Construir un script que reciba una URL, envíe una solicitud HTTP y muestre en consola un análisis básico de las cabeceras devueltas por el servidor.

Requisitos técnicos

- Python 3.x
 - Librería `requests` (puede instalarse con `pip install requests`)
 - Acceso a internet
-

Instrucciones

Crea un archivo llamado `header_analyzer.py`.

Importa el módulo `requests` al inicio del script:

```
import requests
```

Define una función `analyze_headers(url)` que:

- Envíe una petición GET a la URL.
- Imprima todas las cabeceras recibidas en forma de lista.

Añade lógica para:

- Verificar si la URL comienza con `http://` o `https://`. Si no, agregar `http://` por defecto.
- Mostrar advertencia si el servidor devuelve una respuesta diferente a 200.

Dentro de las cabeceras, identifica e imprime en forma destacada si están presentes:

- `Server` → Tecnología del servidor
- `X-Powered-By` → Lenguaje/plataforma (por ejemplo, PHP, ASP.NET)
- `Content-Type`
- `Set-Cookie`
- `Strict-Transport-Security` (si aplica)

Solicita la URL al usuario mediante `input()`.

Opcional

Agrega colores con el módulo `colorama` (solo si ya ha sido usado antes) o imprime las cabeceras destacadas con símbolos para una mejor visualización.

Ejemplo de salida esperada

🔍 Analizando cabeceras para: <http://scanme.nmap.org>

✅ Estado de respuesta: 200 OK

📦 Server: Apache

🔧 X-Powered-By: PHP/7.4.1

📄 Content-Type: text/html

🍪 Set-Cookie: sessionid=abc123; HttpOnly

🔒 Strict-Transport-Security: No presente

🔍 Análisis completo. Usa esta información con fines educativos.

✅ Solución: `header_analyzer.py`

```
import requests
```

```
def analyze_headers(url):
```

```
    # Verifica que la URL tenga un esquema válido
```

```
    if not url.startswith("http://") and not url.startswith("https://"):
```

```
        url = "http://" + url
```

```
    print(f"🔍 Analizando cabeceras para: {url}\n")
```

```
    try:
```

```
        response = requests.get(url, timeout=5)
```

```
        print(f"✅ Estado de respuesta: {response.status_code} {response.reason}\n")
```

```
        headers = response.headers
```

```
        print(f"📦 Cabeceras encontradas:\n")
```

```
        for key, value in headers.items():
```

```
            print(f" - {key}: {value}")
```

```
        print(f"\n📌 Cabeceras destacadas:")
```

```
        print(f"📦 Server: {headers.get('Server', 'No presente')}")
```

```
        print(f"🔧 X-Powered-By: {headers.get('X-Powered-By', 'No presente')}")
```

```
        print(f"📄 Content-Type: {headers.get('Content-Type', 'No presente')}")
```

```
        print(f"🍪 Set-Cookie: {headers.get('Set-Cookie', 'No presente')}")
```

```
        print(f"🔒 Strict-Transport-Security: {headers.get('Strict-Transport-Security', 'No presente')}")
```

```
    except requests.exceptions.MissingSchema:
```

```
        print(f"❌ Error: La URL ingresada no es válida.")
```

```
    except requests.exceptions.ConnectionError:
```

```
        print(f"❌ Error: No se pudo establecer conexión con el servidor.")
```

```
    except requests.exceptions.Timeout:
```

```
print("❌ Error: El servidor tardó demasiado en responder.")
except Exception as e:
    print(f"❌ Error inesperado: {e}")

print("\n🔍 Análisis finalizado. Usa esta información con fines educativos.")

# Programa principal
if __name__ == "__main__":
    print("🔧 Analizador de Cabeceras HTTP - Python y Ciberseguridad")
    target_url = input("👉 Ingresa la URL a analizar (ej: scanme.nmap.org): ")
    analyze_headers(target_url)
```

📦 Ejemplo de ejecución

🔧 Analizador de Cabeceras HTTP - Python y Ciberseguridad
👉 Ingresa la URL a analizar (ej: scanme.nmap.org): scanme.nmap.org

🔍 Analizando cabeceras para: http://scanme.nmap.org

✅ Estado de respuesta: 200 OK

📦 Cabeceras encontradas:

- Server: Apache
- Content-Type: text/html
- Set-Cookie: sessionid=abc123; HttpOnly

📌 Cabeceras destacadas:

- 📦 Server: Apache
- 🔧 X-Powered-By: No presente
- 📄 Content-Type: text/html
- 🍪 Set-Cookie: sessionid=abc123; HttpOnly
- 🔒 Strict-Transport-Security: No presente

🔍 Análisis finalizado. Usa esta información con fines educativos.
