WebScan

# Açıklama

Webscan statik ve dinamik analiz uygulamaları tek bir çatı altında birleştiren bir web güvenlik açığı tespit aracıdır. Statik analiz için Semgrep, dinamik analiz için ise OWASP Zap, NMAP ve SQLMap kullanır. Yaptığı analizin sonucunu ise LaTeX kullanarak pdf formatında rapora dönüştürür.

**Yasal uyarı:** WebScan'in, karşılıklı önceden verilmiş rıza olmadan hedeflere saldırı amaçlı kullanılması yasadışıdır. Tüm geçerli yerel, eyalet ve federal yasalara uymak, son kullanıcının sorumluluğundadır. Geliştiriciler, bu programın kötüye kullanımı veya yol açabileceği hasarlardan dolayı herhangi bir sorumluluk kabul etmemektedir.

# Kurulum Talimatları

## Github

İlk adımda <https://github.com/istec-iuc/WebScan> github reposundan dosyaları pull requestle indirmek gerekiyor.

## Python3

WebScan kodu Python 3.13.0 versiyonunda çalışmaktadır.   
Official website: <https://www.python.org/downloads/release/python-3130/>

Düzgün çalıştığından emin olmak için Python’un pathini Windows 11 için Sistem ve Ortam Değişkenlerine eklemeliyiz;

1. Adım

Başlat menüsünden sistem ve ortam değişkenlerine girin.

A screenshot of a computer

Description automatically generated

1. Adım

Ortam Değişkenleri butonuna tıklayın.

A screenshot of a computer

Description automatically generated

1. Adım

Sistem değişkenlerinin altında Path kısmına çift tıklayın ve açılan pencerede Yeni butonuna tıklayarak Python’un yüklü olduğu dizini ekleyin.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

1. Adım

Düzgün yüklendiğinden ve path’e eklendiğinden emin olmak için cmd’de

‘python –-version’ kodunu çalıştırın ve kontrol edin.

## Playwright

Playwright’ı asenkron bir şekilde hedef websitesinden source dosyaları çekerken kullanıyoruz. Bu işlemi asenkron şekilde yapmamızın sebebi darboğazın önüne geçmek. Her dosyanın indirilmesini beklemek yerine asenkron olarak birden fazla dosya indirebiliyoruz.

Kurulum:

Visual Studio 22 de python enviromentin içerisindeki PowerShell’den veya cmd’den ‘pip install playwright’ kodunu çalıştırmak.



Version 1.48.0

Docs:

https://playwright.dev/docs/intro

## Beautifulsoup4

Beautifulsoup’u hedef websitesinin içeriğini parse’lamak için kullanıyoruz.

Kurulum:

Visual Studio 22 de python enviromentin içerisindeki PowerShell’den veya cmd’den ‘pip install beautifulsoup’ kodunu çalıştırmak.

Docs:

<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

## Aiohttp

Sistemimizle hedef url arasında bağlantı oluşturarak içeriği sistemimize kaydetmeyi sağlar. Asenkron çalışır ve tek session kullanarak performansı arttırır.

Kurulum:

Visual Studio 22 de python enviromentin içerisindeki PowerShell’den veya cmd’den ‘pip install aiohttp’ kodunu çalıştırmak.

Docs:

https://docs.aiohttp.org/en/stable/

## Xmltodict

Xml dosyalarını json’a çevirmeye yarayan bir Python modulü. Nmap outputunu direkt json olarak çıktı alamadığımızdan dolayı nmap outputunu json’a çevirmek için kullanıyoruz.

Kurulum:

Visual Studio 22 de python enviromentin içerisindeki PowerShell’den veya cmd’den ‘pip install xmltodict’ kodunu çalıştırmak.

Docs:

https://pypi.org/project/xmltodict/

## Semgrep

Semgrep’i sistemimize targetten çektiğimiz source dosyalarını analiz etmek için kullanıyoruz.

Semgrep’i windowsta kullanabilmek için wsl(Windows Subsystem for Linux) in kurulu olması gerekmektedir!

Python’da yaptığımız gibi path’i sistem ortam değişkenlerine ekleyin.

Kurulum:

1. Adım

Visual Studio 22 de python enviromentin içerisindeki PowerShell’den veya cmd’den ‘pip install wsl’ kodunu çalıştırmak.

A screen shot of a computer

Description automatically generated  
 WSL Version: 2.3.24.0

2. Adım

Cmd’de wsl’i açıyoruz;



Pipx i wsl’e indiriyoruz;

sudo apt install pipx

Pipx kullanarak semgrep’i wsl’e yüklüyoruz;

pipx install semgrep

1. Adım

Semgrep pathini wsl’e ekliyoruz;

pipx ensurepath

Semgrep Düzgün yüklenmiş mi diye kontrol ediyoruz;

semgrep –-version



Wsl’de semgrep pathini kontrol etmek için;

which semgrep

Docs:

https://semgrep.dev/docs/getting-started/quickstart

semgrep error codes: <https://semgrep.dev/docs/cli-reference#exit-codes>

semgrep local scan: https://semgrep.dev/docs/getting-started/cli

## OWASP Zap

Zap’ı hedef url’e dinamik analiz yapmak için kullanıyoruz. Analiz çıktısı olan JSON dosyasını ise parselayıp LaTeX raporuna uygun formatta yazdırıyoruz.

Requirements:

Java 11 veya daha güncel.

https://www.oracle.com/tr/java/technologies/downloads/#jdk23-windows

Kurulum:

Official websitesinden 2.15.0 sürümünü indiriyoruz;

https://www.zaproxy.org/download/

zap’ı yüklediğimiz klasör önemli çünkü çalıştırmak için commendde cd ile zap.bat’in bulunduğu path’e gitmemiz gerekiyor. WebScanner’ın şu anki halinde zap.bat’in path’i ‘C:/Program Files/ZAP/Zed Attack Proxy’ eğer farklı bir path’e kurulum yaparsanız main fonksiyonun içerisinde   
zap\_dir = "C:/Program Files/ZAP/Zed Attack Proxy"  
yolunu kurduğunuz yola uygun olacak şekilde düzenleyin.

Python’da yaptığımız gibi path’i sistem ortam değişkenlerine ekleyin.

Docs:

zap docs: https://www.zaproxy.org/docs/

Zap cli docs: <https://www.zaproxy.org/docs/desktop/cmdline/>

## Nmap

Nmap’i hedef url’i scanlemek ve açık portları bulup raporlamak için kullanıyoruz.

Kurulum:

Official websitesinden 7.95 sürümünü indiriyoruz.

<https://nmap.org/download.html#windows>

Python’da yaptığımız gibi path’i sistem ortam değişkenlerine ekleyin.

Docs:

https://nmap.org/book/man.html

## SQLmap

SQLmap’i sql injection açıkları otomatik olarak tespit etmek ve raporlamak için kullanıyoruz.

SQLmap’i yüklediğimiz klasör önemli çünkü çalıştırmak için commendde cd ile sqlmap.py’nin bulunduğu path’e gitmemiz gerekiyor. WebScanner’ın şu anki halinde sqlmap.py’nin path’i ‘C:/Users/Administrator/Programs/sqlmap-dev/’ eğer farklı bir path’e kurulum yaparsanız main fonksiyonun içerisinde   
sql\_dir = " C:/Users/Administrator/Programs/sqlmap-dev/"  
yolunu kurduğunuz yola uygun olacak şekilde düzenleyin.

Kurulum:

SQLmap’in github reposundan klonlayarak indirebilirsiniz.

Git clone –depth 1 <https://github>.com/sqlmapproject/sqlmap.git sqlmap-dev

Python’da yaptığımız gibi path’i sistem ortam değişkenlerine ekleyin.

Docs:

<https://github.com/sqlmapproject/sqlmap/blob/master/doc>

## LaTeX

LaTeX’i statik ve dinamik analiz sonuçlarımızı pdf formatında raporlamak için kullanıyoruz.

Kurulum:

MiKTeX’i official sitesinden indirin;  
<https://miktex.org/download>

Version: 24.1

Docs:

https://devdocs.io/latex/  
 <https://www.latex-project.org/help/documentation/>

# Kullanım

## fetch\_all\_links

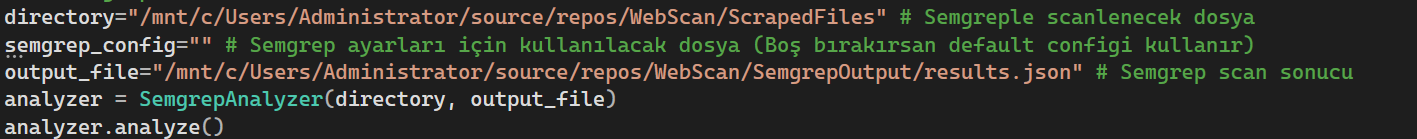
Hedef url’in ve bütün alt sayfalarının source dosyalarını indirebilmek için bu fonksiyonu sonrasında ise fetch\_page fonksiyonunu kullanıyoruz. Bu fonksiyon için hedef url’i belirlemeliyiz. Örnek kullanım;  
A screen shot of a computer program

Description automatically generated

## Semgrep

Semgrep class dizini: C:\Users\Administrator\source\repos\WebScan\semgrep\_analyze.py

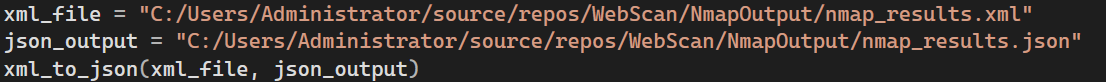
Semgrep’i wsl üzerinden kullandığımız için dosya yollarını örnekte olduğu şekilde girmeliyiz.



## Nmap

Nmap class dizini: C:\Users\Administrator\source\repos\WebScan\nmap\_scanner

Nmap outputu xml olarak verdiği için bu çıktıyı json’a dönüştürmemiz gerekiyor. Du dönüştürme işlemi için:



Nmap\_scanner.py içerisinde 4 çeşit taratma yöntemi bulunmaktadır bunlar sırayıla basic, aggresive, vuln, custom. Raporlama işlemi basic scan’e uygun olacak şekilde tasarlanmıştır diğer taratma yöntemleri raporlarken düzgün çalışmayabilir.

## Zap

Zap class dizini: C:\Users\Administrator\source\repos\WebScan\zap\_scanner

Zap’i kullanırken zap’in yüklü olduğu dizini de girmelisiniz.

Örnek;

A screen shot of a computer

Description automatically generated

## SQLmap

Sql class dizini: C:\Users\Administrator\source\repos\WebScan\SQLmap.py

SQLmap kullanırken sqlmap’in yüklü olduğu dizini de girmelisiniz.

Örnek;

A screen shot of a computer

Description automatically generated

SQLmap.py içerisinde quick ve full sqlmap kullanırken additional optionsla " --dbs, --tables, --colums, --dump, --level=(1 to 5), --risk=(1 to 3)” gibi taratma seçeneklerini commande ekleyebilirsiniz.

## LaTeX

Python kodunda LaTeX kullanırken bazı özel karakterlerin doğru şekilde görünmesi için ek karakterlerle kaçış yapılması gereklidir.

**LaTeX ve Python’da Karakter Kaçışları**

1. **Çift Ters Eğik Çizgi (\\)**: LaTeX’te tek bir ters eğik çizgi \ bir komutu belirtmek için kullanılır (örneğin, \textbf kalın yazı için). Ancak, Python'da tek bir ters eğik çizgi kaçış karakteri olarak işlev görür (örneğin, \n yeni satır için). Python'un bunu kaçış karakteri olarak algılamasını önlemek ve LaTeX’e aynen gönderebilmek için çift ters eğik çizgi \\ kullanırız. Bu, Python'a bu karakteri normal bir ters eğik çizgi olarak işlemesi gerektiğini belirtir.
2. **Çift Süslü Parantez ({{ ve }})**: Süslü parantezler {} LaTeX’te öğeleri gruplamak veya komutların parametrelerini tanımlamak için kullanılır. Ancak, Python’un formatlı yazı tiplerinde (f-string gibi) {} aynı zamanda değişkenlerin yerleştirilmesi için kullanılır. Python’un formatlama sözdizimi ile karışıklık oluşmasını engellemek için {{ ve }} kullanarak süslü parantezleri kaçırırız. Böylece Python bu süslü parantezleri LaTeX’e tekli olarak gönderir.

Örnek:

f"\\multicolumn{{2}}{{|c|}}{{\\textbf{{Vulnerability {idx}}}}} \\\\"

komutu tex dosyasının içerisine

\multicolumn{2}{|c|}{\textbf{Vulnerability 1}} \\  
olarak yazılmaktadır.

LaTeX kullanarak semgrep, zap, sqlmap ve nmap outputlarını parselayıp tek bir raporda topluyoruz parselama ve raporlama işlemi için json\_parser.py kullanıyoruz.  
  
Class dizini: C:\Users\Administrator\source\repos\WebScan\json\_parser

### ZAPParser

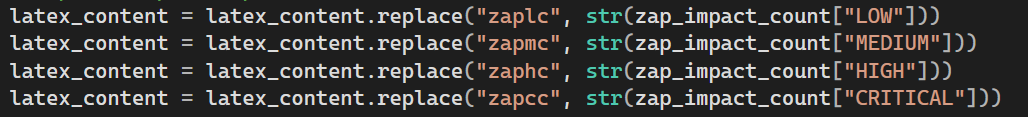
Bu class’ı zap output dosyasını parselamak için kullanıyoruz. Parseladığımız datalar zap\_report içerisinde update\_tex\_report fonksiyonuna aktarılıyor ve bu fonksiyonla

C:\Users\Administrator\source\repos\WebScan\LatexReport.tex dosyasını içerisine yazılıyor.

Bu tex dosyasını açarak hem içeriği görebilir hem de 

Üstteki buton ve ayar kullanılarak pdf’e dönüştürülebilir.

Zap raporunda bulunan risk kodlarını updatelemek için tex dosyasının içerisinde placeholderlar kullanıyoruz. Bu placeholderlar zaplc, zapmc, zaphc, zapcc.



Tex dosyasının içerisinde ise placeholderlar alttaki örnekte olduğu gibi gözükmektedir.

A black background with white text

Description automatically generated

Category altbaşlığının altındaki placeholder:



Vulnerabilities by page altbaşlığının altındaki placeholder:



Not: Semgrep de aynı formatta placeholderları kullanıyor.  
A black background with white text

Description automatically generated





### NmapParser

NmapParser tex raporuna yazdırırken sadece bir placeholder kullanır ve bunun altına parselanmış nmap bilgilerini yazar.



### SQLMapParser

SQLMap text dosyası olarak çıktı verdiği için bu çıktıyı diğerleri gibi json kullanarak parselayamıyoruz. Onun yerine text içerisinden iki“---“ arasında kalan kısmı ayıplayıp tex dosyasına yazdırıyoruz.

SQLMap için tex dosyası içerisindeki placeholder:  
