GOBIERNO DE ESPAÑA
VICEPRESIDENCIA TERCERA DEL GOBIERNO
MINISTERIO DE ASUNTOS ECONÓMICOS Y TRANSFORMACIÓN DIGITAL
SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL

Seminario web: Desarrollo seguro

Ejercicios práctico y de investigación

017 TU AYUDA EN CIBERSEGURIDAD
incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD

ÍNDICE

ÍNDICE DE FIGURAS

1. EJERCICIO 1

La sanitización de variables de entrada es fundamental como vimos en el seminario web. En este caso presentamos la librería “sanitize” que nos va a permitir realizar este trabajo para HTML y directorios en GO.

https://github.com/kennygrant/sanitize

Podremos hacer la sanitización de las siguientes funciones:

Acentos: reemplaza un conjunto de caracteres acentuados con su equivalente ASCII.

```
sanitize.Accents(s string) string
```

Nombres de fichero: normaliza la ruta y nombre de ficheros de forma segura para su uso.

```
sanitize.BaseName(s string) string
```

Reemplazo de entidades HTML: escapa caracteres “<” y “>” para evitar ser interpretado como código HTML por navegadores.

```
sanitize.HTML(s string) string
```

Reemplazo de entidades HTML con filtro: realiza el mismo proceso que el anterior, pero dejando como código los *tags* indicados en los parámetros de la función.

```
sanitize.HTMLAllowing(s string, args...[]string) (string, error)
```

Path: procesa la URL para que sea segura.

```
sanitize.Path(s string) string
```

Como hemos visto anteriormente, gracias a esta librería y los ejemplos anteriores, podemos securizar las entradas.

2. EJERCICIO 2

En este ejercicio de investigación veremos las herramientas más utilizadas para identificar los directorios de una aplicación, cómo está construida y, finalmente, cómo se realizaría una validación de entradas con el objeto de obtener errores en la aplicación web. Este tipo de técnicas también son utilizadas para la identificación de vulnerabilidades y también en un *pentesting*.

La obtención de información es fundamental cuando se buscan errores que no son poco profundos. Algunas herramientas que nos ayudarán de manera rápida y como extensión de Chrome son:

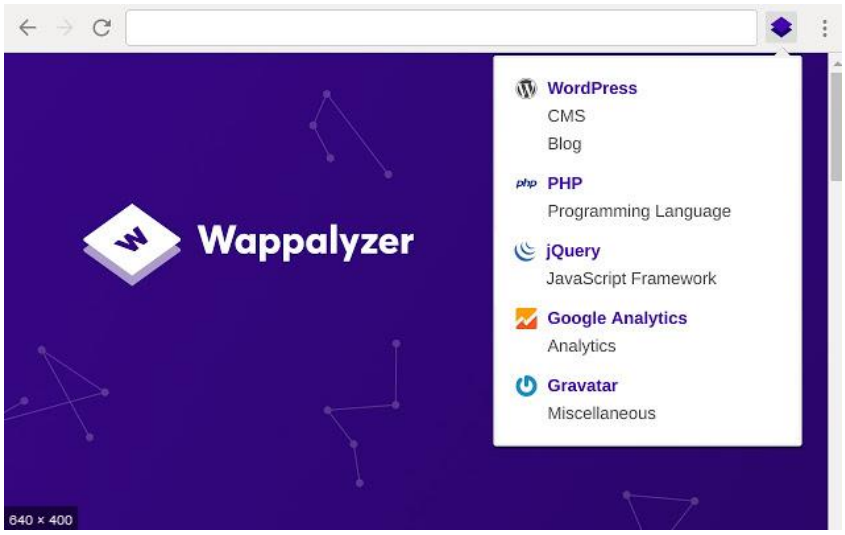

Figura 1 Wappalyzer

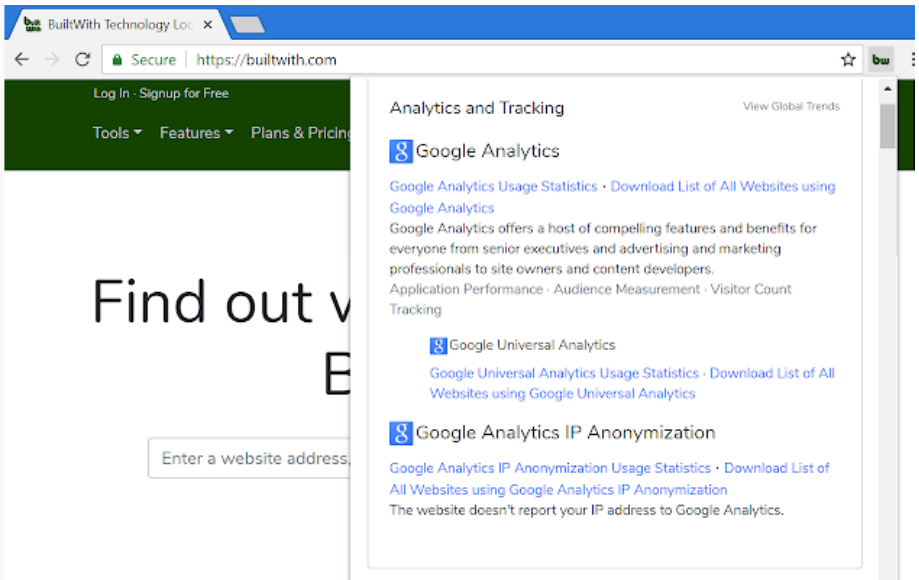

Figura 2 BuiltWith Technology Profiler

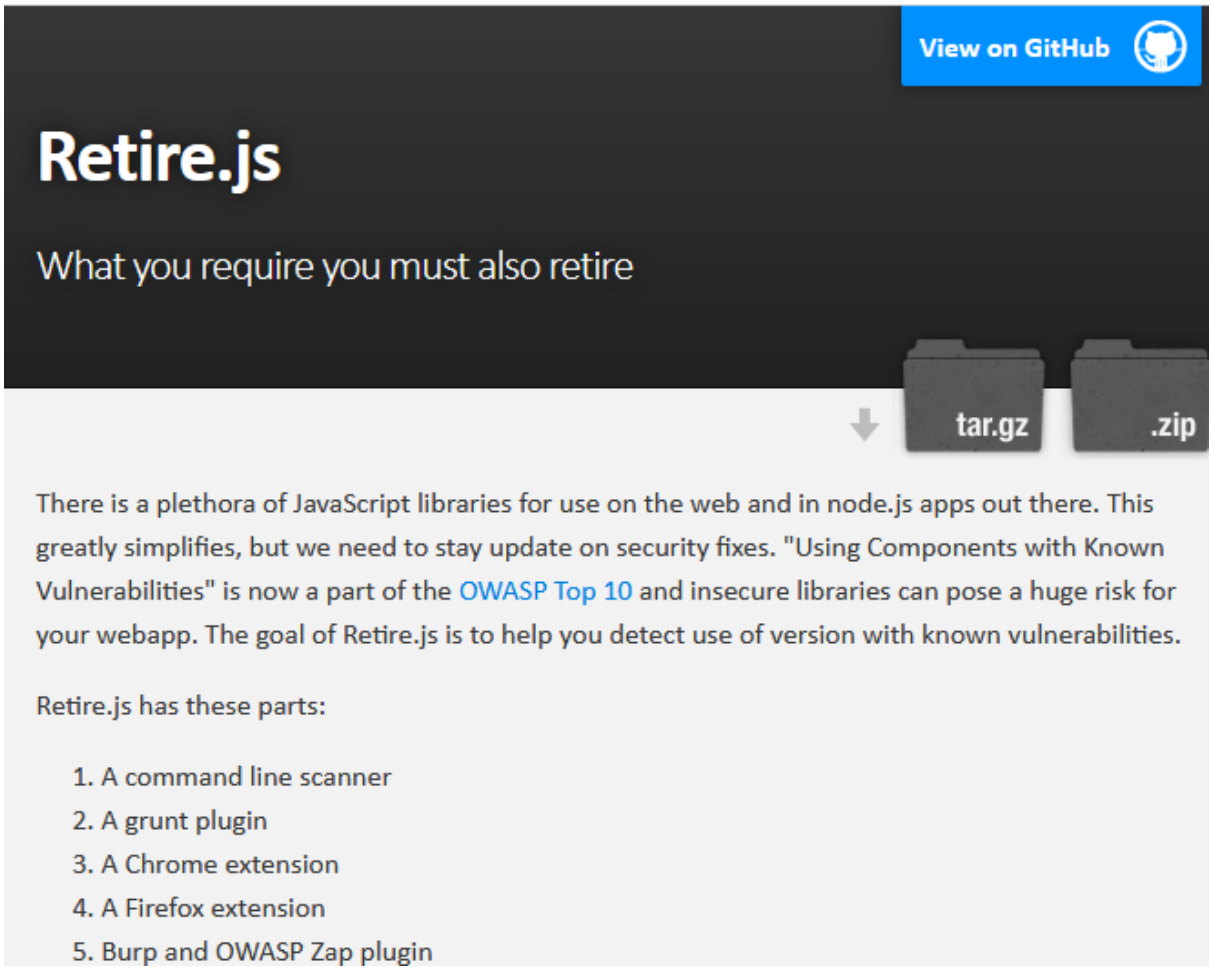

Figura 3 Retire.js

Las extensiones anteriores son solo algunos de los métodos que puede utilizar para obtener información y realizar un perfilado correcto de la aplicación web.

Si a la hora de identificar la aplicación web se obtiene un CMS (Content Management System), se pueden utilizar herramientas específicas:

Figura 4 CMSmap

Figura 5 WPScan

Muchas veces es un arte identificar las rutas donde se encuentran las aplicaciones corriendo y en especial sus directorios. Para ello, se puede utilizar la herramienta Linkclump para facilitar esta labor.

Figura 6 Linkclump

Otro tipo de herramienta que la comunidad utiliza para descubrir directorios sería Dirbuster.

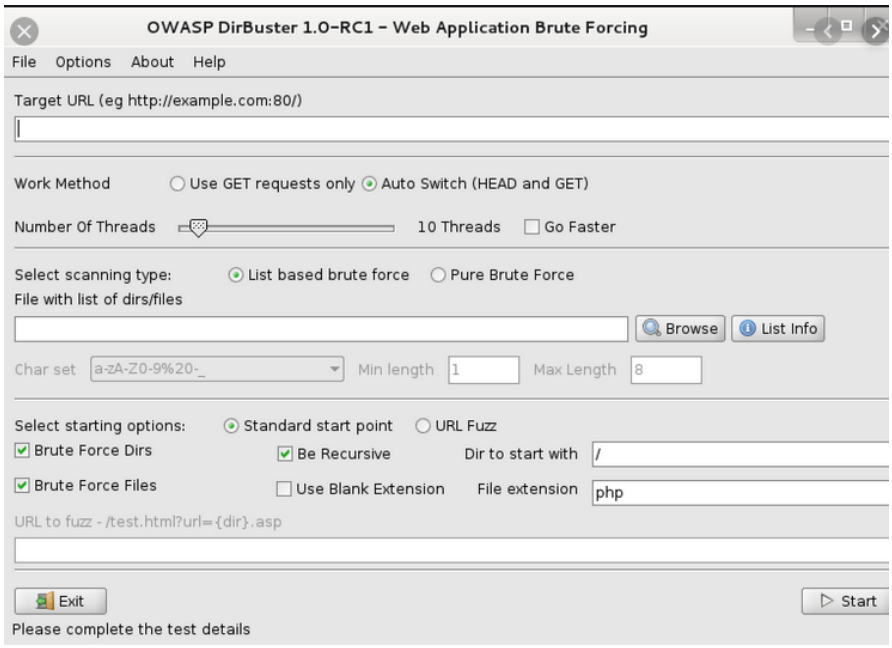

Figura 7 Dirbuster

Por último, cuando tengamos correctamente identificada la aplicación y el directorio de la aplicación, se podría analizar las variables de entrada a la misma mediante POST o GET.

Gracias a BurpSuite y su módulo Intruder se podría realizar este tipo de acciones:

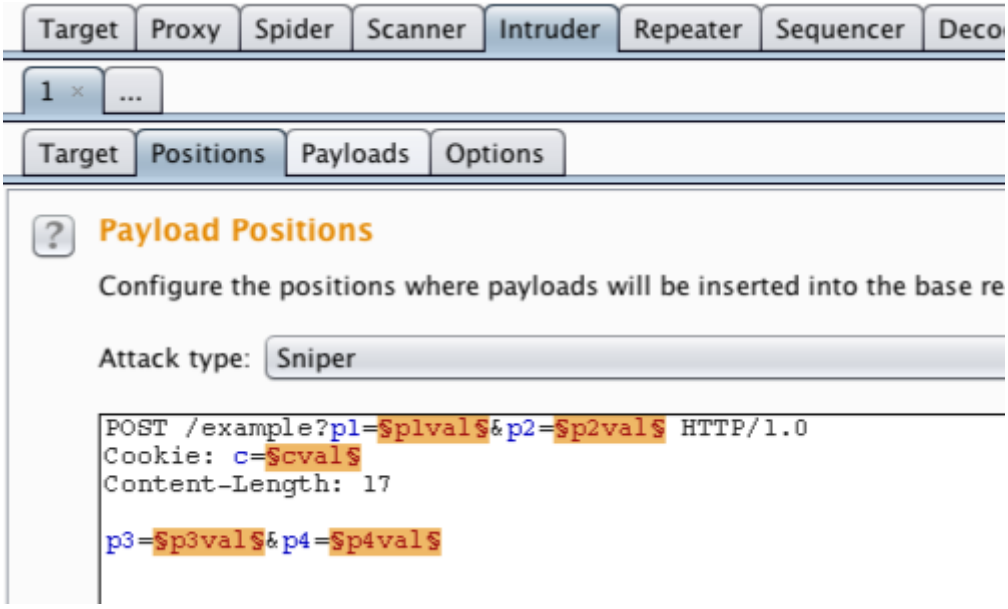

Figura 8 BurpSuite

Dentro del siguiente enlace podrás encontrar un pequeño manual sobre cómo proceder para utilizar el módulo Intruder, así como una web para realizar las pruebas de *testing.* http://testphp.vulnweb.com/login.php