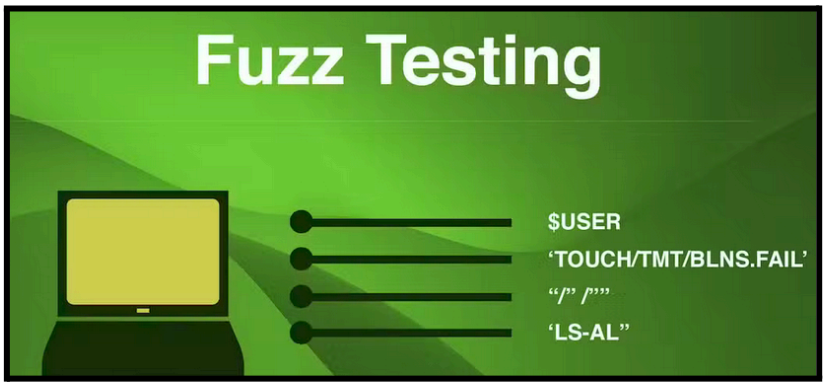
Guía: Fuzzing de Directorios y Subdominios

Esta guía proporciona una explicación clara sobre dos técnicas fundamentales en la enumeración web: fuzzing de directorios y fuzzing de subdominios. Ambas técnicas son esenciales para descubrir recursos ocultos en sitios web, ya sean rutas internas o subdominios relacionados con un dominio principal. El fuzzing se utiliza comúnmente en pruebas de seguridad para identificar posibles áreas vulnerables o servicios no documentados.



Conceptos clave en el Fuzzing

Diccionarios: Listas de palabras predefinidas que se utilizan para probar posibles rutas o subdominios. Estas listas contienen nombres comunes de directorios o subdominios que las herramientas prueban.

Ejemplo: /admin, /login, dev.example.com, test.example.com.

Solicitudes HTTP: Las herramientas de fuzzing envían solicitudes HTTP masivas a las URLs generadas para verificar si alguna devuelve una respuesta válida (por ejemplo, un código de estado 200).

Filtrado de resultados: Durante el fuzzing, es importante filtrar las respuestas irrelevantes (como errores 404) y centrarse en los resultados que devuelvan respuestas interesantes, como páginas de administración.

¿Qué es el Fuzzing de Directorios?

El fuzzing de directorios es el proceso de búsqueda de rutas internas dentro de un sitio web. Se trata de explorar directorios y archivos que no están visibles ni documentados públicamente. Este método es útil para descubrir páginas o recursos ocultos que podrían ser importantes en un análisis de seguridad.

Objetivo: Encontrar directorios o archivos ocultos en un servidor web que puedan contener:

- Páginas de administración (/admin)
- Archivos de configuración (/config.php)
- Zonas de carga de archivos (/uploads)

Ejemplo:

Probar rutas como:

- http://example.com/admin
- http://example.com/uploads

¿Qué es el Fuzzing de Subdominios?

El fuzzing de subdominios es el proceso de búsqueda de subdominios de un dominio principal. Un subdominio es una extensión de un dominio que puede ser usado para alojar servicios o aplicaciones web separadas. Es útil para descubrir otros servidores o aplicaciones que podrían estar ocultos bajo el mismo dominio.

Objetivo: Descubrir otros servidores o aplicaciones dentro de un dominio que podrían estar ocultos, como:

- admin.example.com
- dev.example.com

Ejemplo:

Probar subdominios como:

- http://dev.example.com
- http://test.example.com

Práctica recomendada

Ejercicio de Fuzzing de Directorios:

Utilizar Gobuster para buscar directorios ocultos en un sitio web.

```
# gobuster dir -u http://example.com -w /usr/share/wordlists/dirb/common.txt
```

Intentar con Feroxbuster para buscar rutas y archivos.

```
# feroxbuster -u http://example.com -w /usr/share/wordlists/dirb/common.txt
```

Ejercicio de Fuzzing de Subdominios:

Utilizar Gobuster DNS para buscar subdominios de un dominio.

```
# gobuster dns -d example.com -w
/usr/share/seclists/Discovery/DNS/subdomains-top1million-110000.txt
```

Probar con wfuzz para fuzzing en el encabezado Host.

```
# wfuzz -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-110000.txt -u
http://exam
```

Gobuster

Gobuster es una herramienta de fuerza bruta utilizada para descubrir directorios, archivos, subdominios y Virtual Hosts en aplicaciones web. Se utiliza para realizar ataques de reconocimiento y enumeración durante pruebas de penetración.

Dir:
`gobuster dir -u http://example.com -w /ruta/a/wordlist.txt`
Busca directorios y archivos en la URL o dominio especificado usando una lista de palabras (wordlist).

Vhost:
`gobuster vhost -u http://example.com -w /ruta/a/wordlist.txt`
Busca Virtual Hosts en el dominio especificado.

DNS:
`gobuster dns -d example.com -w /ruta/a/wordlist.txt`
Realiza una búsqueda de subdominios para el dominio especificado.

Fuzz:
`gobuster fuzz -u http://example.com/FUZZ -w /ruta/a/wordlist.txt`
Realiza un ataque de fuzzing en la URL especificada, reemplazando el término `FUZZ` con palabras de la lista de palabras.

Opciones de gobuster:

-u: URL o dominio de destino.
Ejemplo: `-u http://example.com`

-w: Ruta al archivo de wordlist.
Ejemplo: `-w /ruta/a/wordlist.txt`

-t: Número de hilos (threads).
Ejemplo: `-t 50`

Los hilos son el número de solicitudes que Gobuster envía al mismo tiempo.

- Más hilos = más rápido, pero más carga en el servidor.
- Menos hilos = más lento, pero menos carga.

Opciones de gobuster:

-x: Extensiones de archivos (como .php, .html).
Ejemplo: `-x .php,.html`

-l: Limitar la profundidad (niveles de directorio).
Ejemplo: `-l 2`

-o: Guardar salida en archivo.
Ejemplo: `-o resultado.txt`

-k: Ignorar SSL (certificados no válidos).
Ejemplo: `-k`

WFuzz

WFuzz es una herramienta poderosa y flexible para realizar fuzzing, especialmente útil para encontrar directorios, archivos y subdominios en servidores web. Aquí te explicaré cómo usar WFuzz para estos tres escenarios comunes.

1. Fuzzing de Directorios y Archivos

WFuzz permite realizar fuzzing para descubrir directorios y archivos ocultos en un sitio web. El proceso básico es enviar solicitudes HTTP con un diccionario de palabras (wordlist) y observar las respuestas para identificar recursos disponibles.
Comando Básico para Fuzzing de Directorios y Archivos:

```
# wfuzz -w /path/to/wordlist.txt -u "http://example.com/FUZZ" -t 50
```

- -w: Define la ruta al archivo de lista de palabras (wordlist).
- -u: La URL de destino. La palabra FUZZ se reemplaza por las entradas de la lista.
- -t: Define el número de hilos concurrentes. En este caso, 50 hilos.

Ejemplo de Búsqueda de Directorios:

```
# wfuzz -w /usr/share/seclists/Discovery/Web-Content/dirb/common.txt -u
"http://example.com/FUZZ/"
```

Esto buscará directorios en el sitio example.com usando una lista de palabras común (como la de dirb).

Ejemplo de Búsqueda de Archivos:

```
# wfuzz -w /usr/share/seclists/Discovery/Web-Content/dirb/common.txt -u
"http://example.com/FUZZ.html"
```

Este comando buscará archivos con la extensión .html en el sitio example.com.

2. Fuzzing de Subdominios

El fuzzing de subdominios en solicitudes HTTP es otra tarea común. En este caso, utilizamos el encabezado Host para realizar fuzzing de subdominios.
Comando Básico para Subdominios:

```
# wfuzz -w /path/to/subdomains.txt -u "http://example.com" -H "Host: FUZZ.example.com" -t 50
```

-H "Host: FUZZ.example.com": Este encabezado se utiliza para realizar fuzzing de subdominios. Reemplaza FUZZ con los términos de la lista de palabras.

Ejemplo de Búsqueda de Subdominios:

```
# wfuzz -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt -u "http://example.com" -H "Host: FUZZ.example.com" -t 50
```

Este comando buscará subdominios en example.com usando la lista de palabras subdomains-top1million-5000.txt. Puedes encontrar esta lista en directorios como **/usr/share/seclists/Discovery/DNS/.**

Notas sobre el uso del encabezado Host:

Cuando realices fuzzing de subdominios, WFuzz enviará solicitudes HTTP con el encabezado Host modificado. Esto permite que puedas acceder a subdominios no directamente visibles, pero que están registrados en el DNS.

3. Fuzzing de Directorios y Archivos con Respuestas Específicas

Puedes también realizar fuzzing basado en las respuestas del servidor, por ejemplo, buscar directorios o archivos que devuelvan códigos HTTP específicos como 403 (Forbidden), 404 (Not Found) o 200 (OK).

Comando para Filtrar Respuestas 200 (OK):

```
# wfuzz -w /usr/share/seclists/Discovery/Web-Content/dirb/common.txt -u "http://example.com/FUZZ" -t 50 -mc 200
```

-mc 200: Filtra las respuestas que devuelvan un código 200 OK.

Comando para Filtrar Respuestas 403 (Forbidden):

```
# wfuzz -w /usr/share/seclists/Discovery/Web-Content/dirb/common.txt -u "http://example.com/FUZZ" -t 50 -mc 403
```

Esto buscará directorios o archivos que devuelvan un código 403 Forbidden, lo cual puede indicar que hay recursos ocultos en el sitio.

4. Fuzzing de Parámetros en la URL

Otra técnica útil es realizar fuzzing en los parámetros de la URL. WFuzz permite modificar los parámetros para encontrar vulnerabilidades o recursos ocultos.
Comando para Fuzzing de Parámetros:

```
# wfuzz -w /path/to/wordlist.txt -u "http://example.com/page.php?id=FUZZ" -t 50
```

Esto probará diferentes valores para el parámetro id en la URL.

5. Opciones Avanzadas de WFuzz

- -mc: Filtra por código de respuesta HTTP. Ejemplo: -mc 200 para respuestas exitosas.
- -t: Define el número de hilos concurrentes para aumentar la velocidad del fuzzing.
- -H: Permite añadir encabezados personalizados en la solicitud. Ejemplo: -H "Host: FUZZ.example.com" para subdominios.
- -b: Permite añadir cookies a las solicitudes. Ejemplo: -b "sessionid=abc123".
- --hl: Muestra sólo las respuestas que tengan una longitud mayor que el valor especificado.
- -u: Define la URL objetivo donde se hará el fuzzing.
- --hc: Filtra las respuestas basadas en el tamaño del contenido. Ejemplo: --hc 500 muestra respuestas con más de 500 bytes.