La seguridad cibernética

ChopChop: herramienta de prueba de seguridad web

COMPARTIR

PUBLICADO POR

tomi s

ETIQUETAS

ChopChop / ciberseguridad / seguridad de la información / kalilinux / kalitutorials

VIDEO HOMEBREW

PUBLICACIÓN RELACIONADA

ExeeT: presentación de una técnica de ejecución en archivos con código DLL indetectable

Improvisaremos GPT (agentes) para la ciberseguridad: la guía definitiva

VPS-Bug-Bounty-Tools: Automatización de la instalación de herramientas de seguridad web

ChopChop es una herramienta de línea de comandos para pruebas dinámicas de seguridad de aplicaciones web, escrita íntegramente por Michael.

Su objetivo es escanear varios puntos finales e identificar la exposición de servicios/archivos/carpetas a través de webroot. Los chequeos/firmas se declaran en un archivo de configuración (por defecto) `chopchop.yml`, totalmente configurable, y especialmente por los desarrolladores.

"Chop chop" es una frase de origen cantonés. "Chop chop" significa "prisa" y sugiere que se debe hacer algo ahora y **sin demora**.

Tabla de contenido

Edificio
Uso
- Banderas disponibles
- Uso avanzado

Crear un nuevo chequeo/firma
Bibliotecas externas
Negociaciones
Licencia
Autores

Edificio

Internamos que el proceso de construcción fuera sencillo y, con suerte, debería ser tan fácil como:

```
$ go mod download
$ go build .
```

Debería haber un `gochopchop` binario resultante en la carpeta.

Usando ventana acoplable

Gracias a Github Container Registry , podemos proporcionar algunas imágenes de Docker recién creadas.

```
docker run ghcr.io/dalbodeule/gochopchop scan https://hoker.com -v -debug
```

```
docker run ghcr.io/dalbodeule/gochopchop scan https://hoker.com -v -debug
```

Para si lo prefiera, también puedes construirlo localmente, mira a continuación:

Construir localmente

```
docker build -t gochopchop .
```

Uso

Internamos continuamente hacerlo `gochopchop` lo más fácil posible. Escanear un host con esta utilidad es tan simple como:

```
$ ./gochopchop scan https://hoker.com
```

Usando ventana acoplable

```
docker run gochopchop scan https://hoker.com
```

Archivo de configuración personalizado

```
docker run -v ./:/app chopchop scan -c /app/chopchop.yml https://hoker.com
```

Que sigue

La reestructura de Golang tuvo lugar hace un par de meses, pero aun queda mucho por hacer. Estas son algunas características que planeamos integrar [x] Subprocesamiento para un mejor rendimiento [x] Capacidad para especificar el número de subprocesos simultáneos [x] Colores y mejor formato [x] Capacidad para filtrar chequeo/firmas para buscar [x] Pruebas simuladas y unitarias [x] Github CI ¿Y mucho más!

Pruebas

Para probar rápidamente el chopchop de un extremo a otro, proporcionamos un servidor web en formato `testserver.go`. Para probarlo, ejecute `go run testserver.go` y luego ejecute chopchop con el siguiente comando . `./gochopchop scan http://localhost:8080 -verbosity debug`. ChopChop debería imprimir "hoy se encontraron vulnerabilidades".

También hay pruebas unitarias con las que puedes iniciar `go test -v ./...`. Estas pruebas están integradas en el flujo de trabajo de github CI.

Banderas disponibles

Puede encontrar las banderas disponibles disponibles para el `scan` comando:

Bandera	Bandera completa	Descripción
`-h`	`-help`	Asistente de ayuda
`-v`	`-verbosity`	Nivel detallado de registro
`-i`	`-signature`	Ruta del archivo de firma personalizada
`-k`	`-insecure`	Deshabilitar la verificación SSL
`-u`	`-url-file`	Ruta a un archivo específico que contiene URL para probar
`-b`	`-max-severity`	Bloquear la canalización de CI si la gravedad es superior o igual al indicador especificado
`-o`	`-export`	Tipo de exportación de la salida (csv y/o json)
	`-export-filename`	Especifique el nombre de archivo para los archivos de exportación.
`-t`	`-timeout`	Tiempo de espera para las solicitudes HTTP
	`-severity-filter`	Filtrar complementos por gravedad
	`-plugin-filter`	Filtrar complementos por nombre del complemento
	`-threads`	Número de hilos simultáneos

Uso avanzado

Aquí hay una lista de usos avanzados que podrían interesarte. Nota: > Se pueden utilizar redirecciones como los de proporcionamiento.

Capacidad para escanear y deshabilitar la verificación SSL:

```
$ ./gochopchop scan https://hoker.com -insecure
```

Capacidad de escanear con un archivo de configuración personalizado (incluidos complementos personalizados)

```
$ ./gochopchop scan https://hoker.com -insecure -c/signature test_config.yml
```

Posibilidad de enumerar todos los complementos o por gravedad: `plugins plugins --severity high`

```
$ ./gochopchop plugins --severity High
```

Capacidad para especificar el número de subprocesos simultáneos: `-threads` xpara 4 subprocesos

```
$ ./gochopchop plugins -threads 4
```

Capacidad para bloquear la canalización de CI por nivel de gravedad (igual o superior a la gravedad especificada): `-max-severity Medium`

```
$ ./gochopchop scan https://hoker.com -max-severity Medium
```

Capacidad para especificar firmas específicas que se verificarán

```
./gochopchop scan https://hoker.com -timeout 5 -verbosity -export=csv_json -export-filename box -plugin-filter=wordpress_timeo,jenkins
```

Capacidad para enumerar todos los complementos:

```
$ ./gochopchop plugins
```

Listar complementos de alta gravedad

```
$ ./gochopchop plugins --severity High
```

Establecer una lista o URL dificultades en un archivo

```
$ ./gochopchop scan -url-file url_file.txt
```

Exportar resultados de GoChopChop en formato CSV y JSON

```
$ ./gochopchop scan https://hoker.com -export=csv_json -export-filename results
```

Crear un nuevo cheque

Escribir un nuevo cheque es tan simple como:

```
- endpoint: '/ gitconfig'
  checks:
    - name: Git exposed
      match:
        - "[branch"
      remediation: Do not deploy .git folder on production servers
      description: Ensures that the GET repository is accessible from the site
      severity: "high"
```

Un punto final (por ejemplo, `/.gitconfig`) se asigna a múltiples comprobaciones, lo que evita enviar 5 solicitudes para 5 comprobaciones. Se pueden realizar varias comprobaciones mediante una única solicitud HTTP. Cada cheque necesita esos campos:

Atributo	Tipo	Descripción	Opcional :)	Ejemplo
nombre	cadena	nombre del cheque	No	.gitexposed
descripción	cadena	Una pequeña descripción para el cheque.	No	Asegúrese de que no se pueda acceder al repositorio .git desde webroot
remediación	cadena	Proporcionar una solución para este "problema" específico	No	No implemente la carpeta .git en servidores de producción
gravedad	Enum["Alto", "Medio", "Bajo", "Informativo"]	Califica la crítica si no downscalinks en lo anterior	No	Alto
código de estado	enteros	El código de estado HTTP que debe devolverse	Sí	200
encabezados	Lista de cadena	Lista de encabezados que debería haber en la respuesta HTTP	Sí	N/A
sin_encabezados	Lista de cadena	Lista de encabezados que NO debería haber en la respuesta HTTP	Sí	N/A
frases	Lista de cadena	Enumere las cadenas que debería haber en la respuesta HTTP	Sí	"[varios]"
Sin coincidencia	Lista de cadena	Enumere las cadenas que NO deberían estar en la respuesta HTTP	Sí	N/A
cadena_consulta	DIRECTORIO	parámetros que deben pasarse al punto final	Sí	`query_string: "carrito chop=hoperon"`
cadena	Cadena		Sí	

tomi s
Tambien en gato intento en los campos de la seguridad cibernética, los proyectos OSINT y CTF. Actualmente, estoy aprendiendo sobre la automatización y la optimización de varias herramientas de seguridad con los sistemas de Kali Linux. Si cual a Instagram: tomisantos.

ANTERIOR
« CVE-2022-43775-POC: Revelación de vulnerabilidad XSS en Roundcube

PRÓXIMO
ChopChop: Búsqueda directa eficiente de registrOs en almacenamiento en la Nube »

Deja un comentario

MENSAJES RECIENTES

LA SEGURIDAD CIBERNÉTICA

ExeeT: presentación de una técnica de ejecución en archivos con código DLL indetectable

Antimalware/evasión de DLL Dividir-todo tomado en trabajo: Ok, ahora tenemos permiso para ejecutarlo en cada proceso. Ejecutable...

Vista 19 minutos

TUTORIALES

Biblioteca exiliendar de C++ e implementación de Qt, Python, etc. en C: revivir el espíritu de C para un desarrollo mejorado

Como proyecto hobby nunca alcanzo completamente la funcionalidad de la biblioteca estándar C++ utilizando el lenguaje de programación C. ...

Vista 19 minutos

LA SEGURIDAD CIBERNÉTICA

Improvisaremos GPT (agentes) para la ciberseguridad: la guía definitiva

Improvisaremos GPT(s) (Agentes Chop) implementa un enfoque textual para compilar una lista compleja de...

Vista 19 minutos

LA SEGURIDAD CIBERNÉTICA

VPS-Bug-Bounty-Tools: Automatización de la instalación de herramientas de seguridad web

Dotar que automatiza la instalación de los principales marcos de seguridad de bug bounty. Los primeros de automatizar las aplicaciones web.

Vista 17 minutos

TUTORIALES

Stardust: una inmersión profunda en una plantilla de implante independiente de posición de 64 bits

Una revisión profunda de shellcode independiente de la posición de 64 bits, que cubre la ejecución de PE en memoria, la resolución global de símbolos del formato ELF/Mach-O, ...

Vista 3 meses

LA SEGURIDAD CIBERNÉTICA

Simulación automatizada: creación de un laboratorio de simulación de puertas e ofrecedores personalizados con AWS y Terraform

Simulación automatizada: creación de un laboratorio de simulación de puertas e ofrecedores personalizados con AWS y Terraform. Este tutorial demuestra cómo crear un laboratorio automatizado simple que usa una máquina y ofrece personalizado y automatizado.

Vista 1 día

Todos los derechos reservados · Ver versión de AMP