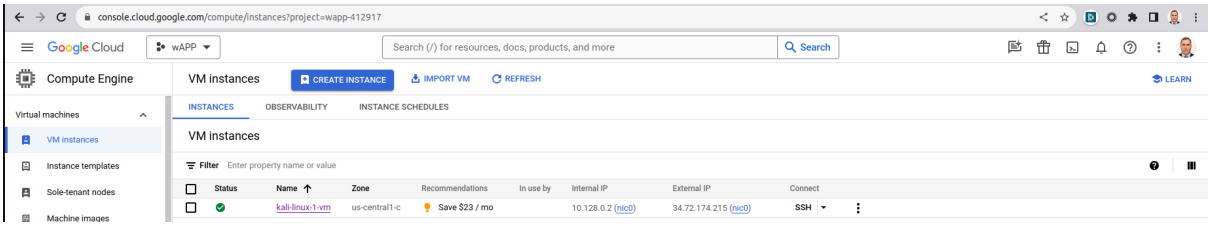
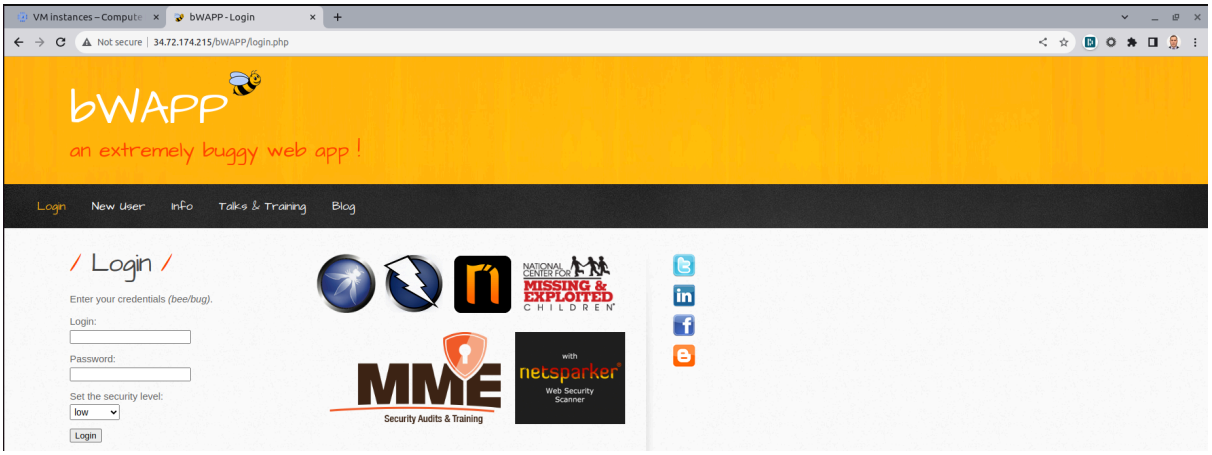
Informe Técnico

Este proyecto tiene como finalidad desplegar una aplicación vulnerable, tener visibilidad de los eventos de seguridad relacionados con dichos servicios, sugerir una mejora en la arquitectura para mitigar las vulnerabilidades, disponibilizar la información recolectada por medio de una API y entregar la información recolectada en éste informe técnico.

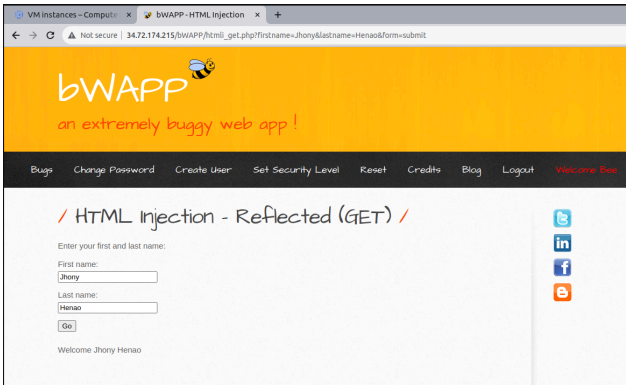
Desarrollo de la Prueba

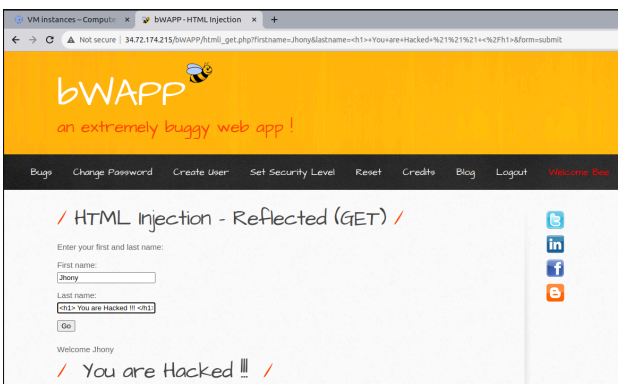
1. La implementación de la aplicación vulnerable se realizó por medio de una VM Instance en la nube de GCP con sistema operativo Linux, la aplicación utilizada se llama bWAPP (https://sourceforge.net/projects/bwapp/).





2. La simulación de ataques se realizó por medio directo de la aplicación tomando dos vectores de ataque diferentes, uno de aplicación con un HTML Injection y una Denegación de Servicio a nivel de red/conectividad (https://github.com/gkbrk/slowloris). Para el punto de visualización de logs se utilizó el WAF de CloudFlare (configurado más adelante).





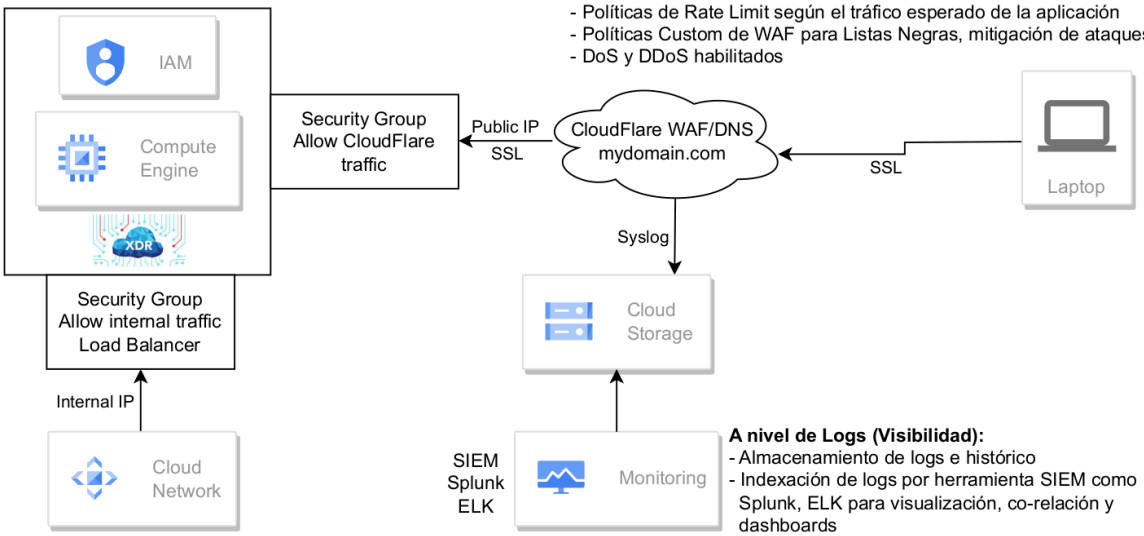
```
jhenao@jah396328:~/Downloads/slowloris-master$ ./slowloris.py -p 80 -v 34.72.174.215
[03-02-2024 15:03:52] Attacking 34.72.174.215 with 150 sockets.
[03-02-2024 15:03:52] Creating sockets...
[03-02-2024 15:03:52] Creating socket nr 0
[03-02-2024 15:03:52] Creating socket nr 1
[03-02-2024 15:03:52] Creating socket nr 2
[03-02-2024 15:03:52] Creating socket nr 3
[03-02-2024 15:03:52] Creating socket nr 4
[03-02-2024 15:03:52] Creating socket nr 5
[03-02-2024 15:03:53] Creating socket nr 6
[03-02-2024 15:03:53] Creating socket nr 7
[03-02-2024 15:03:53] Creating socket nr 8
[03-02-2024 15:03:53] Creating socket nr 9
[03-02-2024 15:03:53] Creating socket nr 10
[03-02-2024 15:03:53] Creating socket nr 11
[03-02-2024 15:03:53] Creating socket nr 12
```

```
[03-02-2024 15:05:20] Creating socket nr 143
[03-02-2024 15:05:20] Creating socket nr 144
[03-02-2024 15:05:20] Creating socket nr 145
[03-02-2024 15:05:20] Creating socket nr 146
[03-02-2024 15:05:20] Creating socket nr 147
[03-02-2024 15:05:20] Creating socket nr 148
[03-02-2024 15:05:21] Creating socket nr 149
[03-02-2024 15:05:21] Sending keep-alive headers...
[03-02-2024 15:05:21] Socket count: 150
[03-02-2024 15:05:21] Sleeping for 15 seconds
[03-02-2024 15:05:36] Sending keep-alive headers...
[03-02-2024 15:05:36] Socket count: 150
[03-02-2024 15:05:36] Sleeping for 15 seconds
[03-02-2024 15:05:51] Sending keep-alive headers...
[03-02-2024 15:05:51] Socket count: 150
[03-02-2024 15:05:51] Creating 82 new sockets...
```

3. Arquitectura propuesta para mitigar ataques

A nivel de VM / Cloud:
- Agente XDR para detectar cualquier evento atípico o sospechoso
- Política de IAM definidas con accesos (Least Privilege)
- FW o Security Groups permitiendo tráfico de red interna o balanceadores y de las IP públicas de CloudFlare (WAF) por su interfaz externa.

A nivel de CloudFlare (WAF):
- Configuración de registros DNS para el acceso público a la aplicación
- Políticas de WAF habilitadas y revisadas según el tipo de aplicación
- Políticas de Rate Limit según el tráfico esperado de la aplicación
- Políticas Custom de WAF para Listas Negras, mitigación de ataques
- DoS y DDoS habilitados

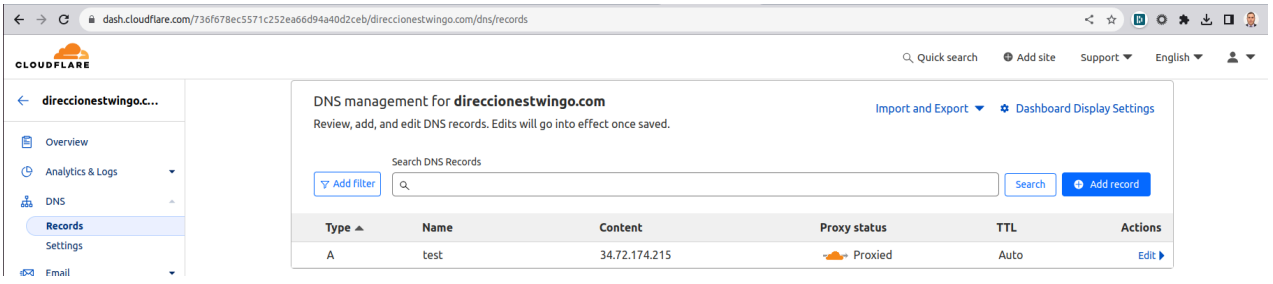


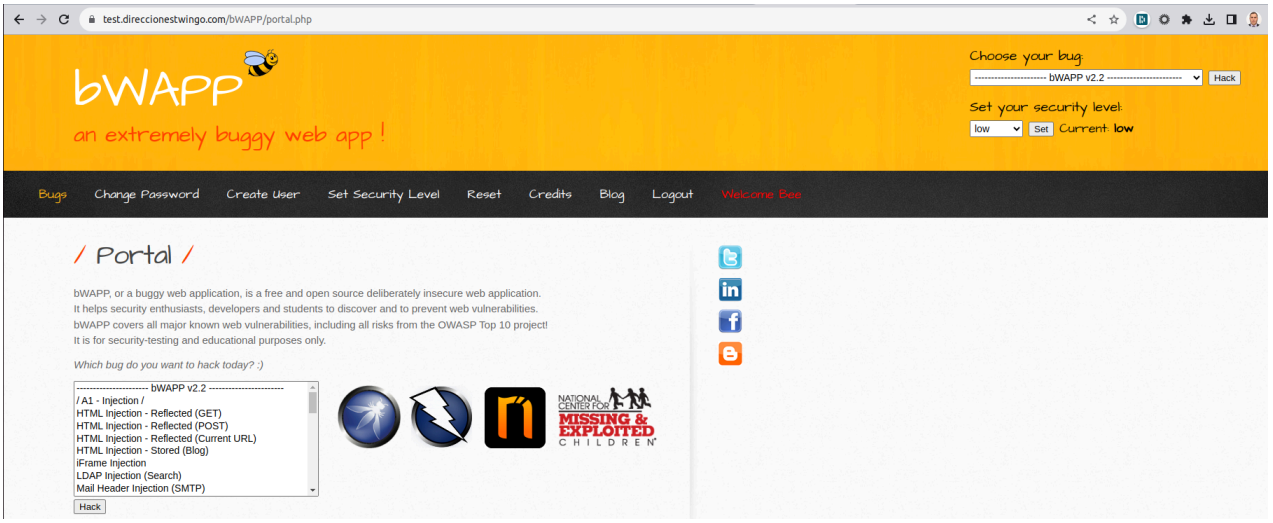
A nivel de Logs (Visibilidad):
- Almacenamiento de logs e histórico
- Indexación de logs por herramienta SIEM como Splunk, ELK para visualización, co-relación y dashboards

4. Herramienta visualización de logs para observar los patrones de los ataques realizados

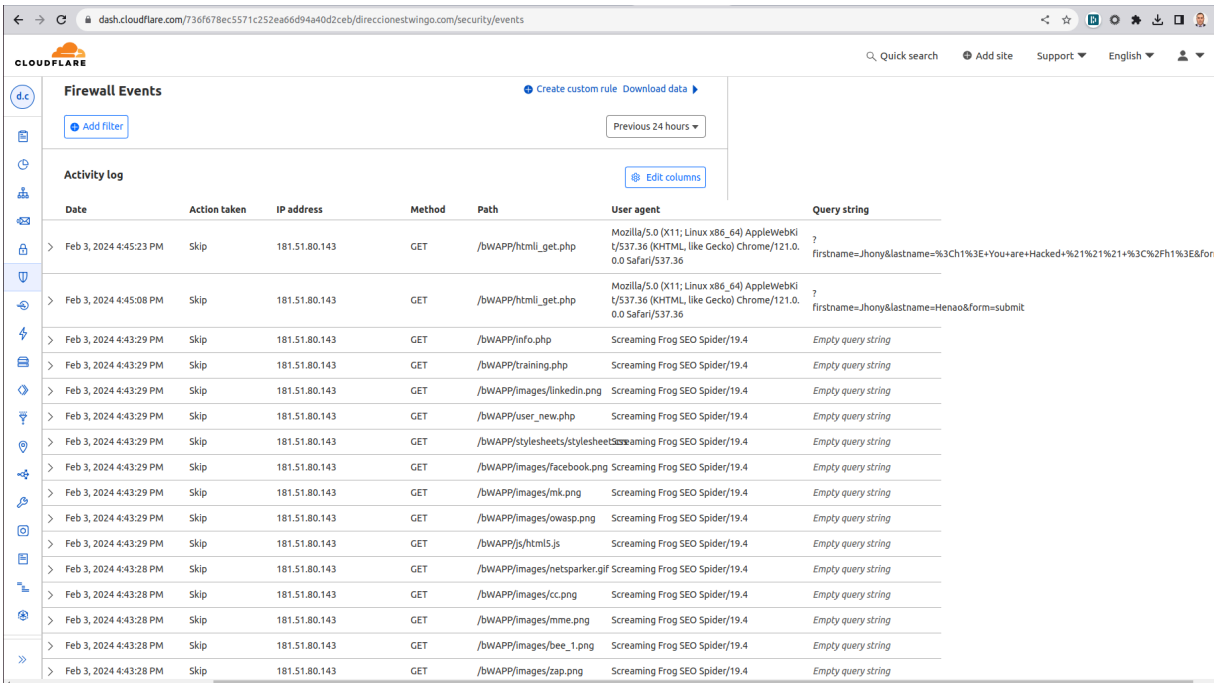
Como herramienta utilicé CloudFlare ya que tiene una versión gratuita que aunque es muy limitada al momento de exportar y trabajar con los logs me dió observabilidad de cada ataque y de cómo se manifestaba a nivel de red y aplicación.

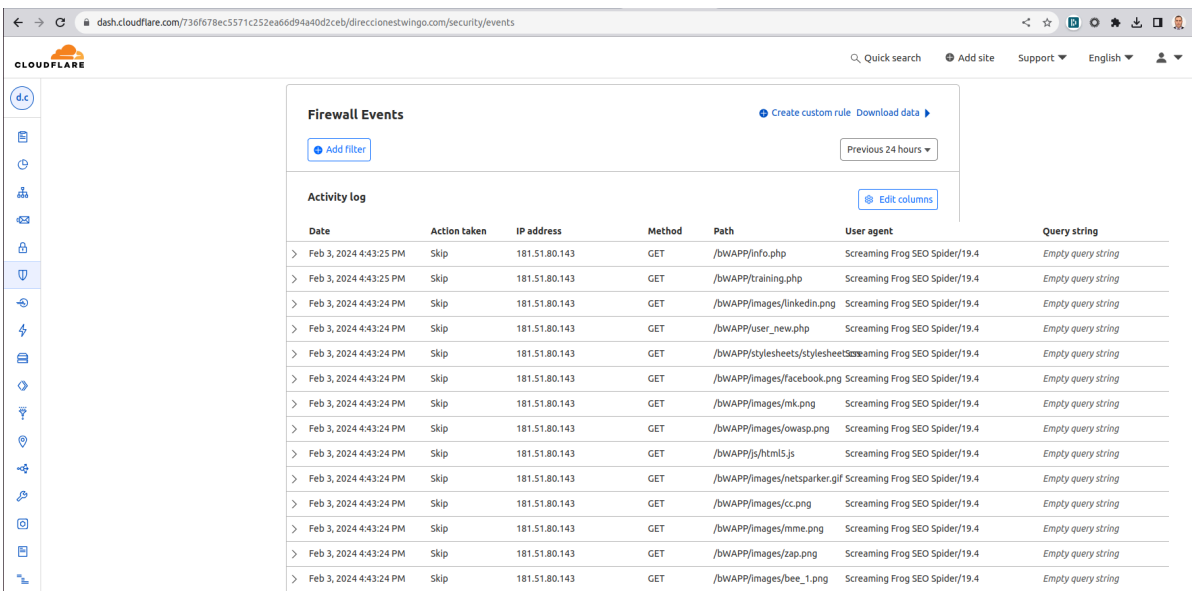
Realicé la inscripción de un dominio en CloudFlare y creé el respectivo registro para alcanzar la aplicación desplegada en GCP.





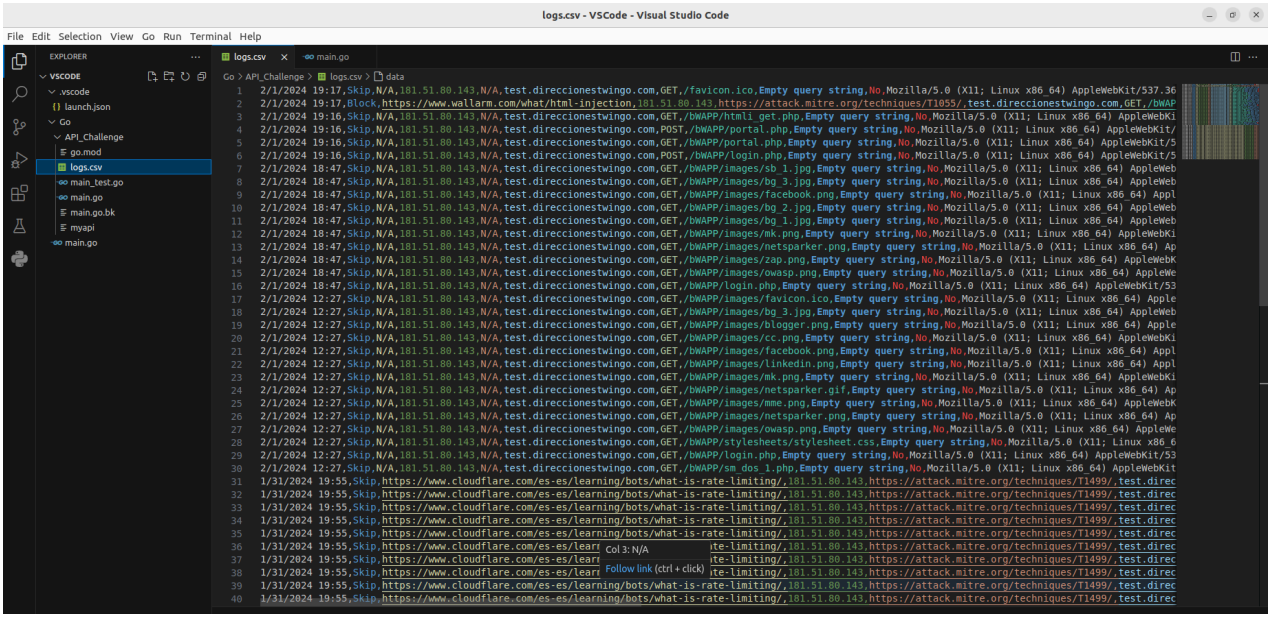
Podemos observar el ataque de DoS que fue simulado con ayuda de una herramienta de SEO Screaming Frog la cual puede realizar numerosas peticiones a todos los subdominios y carpetas publicados en el sitio, además del HTML Injection que se puede observar en el Query string.



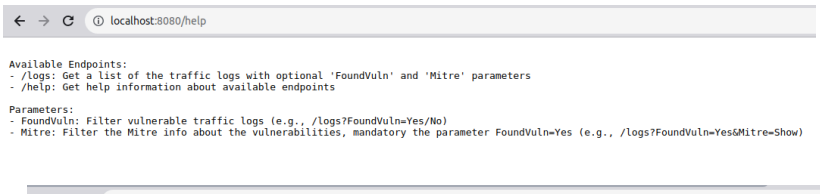


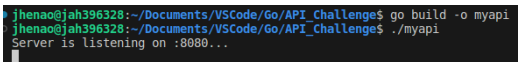
5. Desarrollo de API para visualización de la información recolectada durante el reto

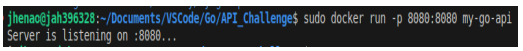
Debido a que la capa gratuita de CloudFlare no me permitió exportar de ningún modo los logs, realicé una extracción manual copiando desde la opción Firewall Events los logs recolectados del ejercicio y haciendo un parseo con ayuda de excel hasta terminar convirtiendo la información a un formato csv que pudiera ser consultable desde la API. Con éste paso lo que hice fue simular el reenvío de logs por syslog que se pudiera hacer hacia cualquier almacenamiento externo.

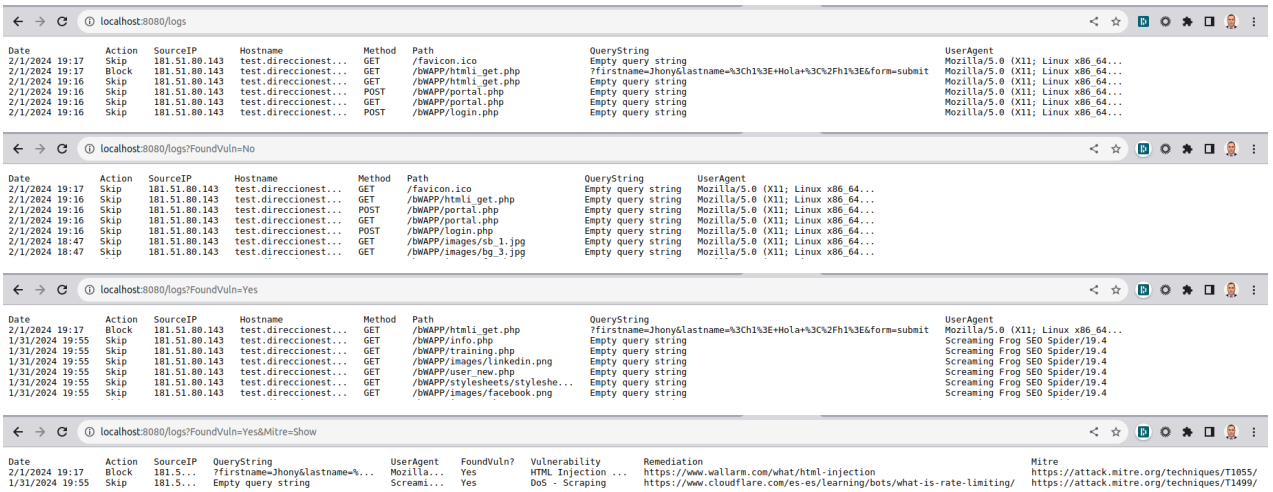


Por medio de la API se pueden visualizar los logs, además de poder filtrar por tráfico con vulnerabilidades con su respectiva posible mitigación y/o documentación. También es posible listar del tráfico con vulnerabilidades su correspondencia con la matriz Mitre, todo ésto es basado en Query params sobre la API.









Bonus y Anexos

- Desplegar API con el lenguaje de programación Golang (Ver archivo main.go anexo)
- Test unitario para el endpoint (Ver archivo main_test.go anexo)
- Desplegar el microservicio en kubernetes (Ver archivos Dockerfile, deployment.yaml y service.yaml)
- Asociar vulnerabilidades encontradas acorde a la matriz Mitre (Ver endpoint http://localhost:8080/logs?FoundVuln=Yes&Mitre=Show en documentación punto 5