Universidad de Talca
Facultad de Ingeniería
Departamento de Ciencias de la Computación

Seguridad Informática

Laboratorio 6

Erik Regla
eregla09@alumnos.utalca.cl

12 de julio de 2020

Índice

1. Actividades

Elegimos como objetivo disponible en la web la pagina principal de la universidad por motivos legales y de pruebas. Además que ya conocemos algo de información al respecto y estas herramientas no necesariamente pueden encontrar información de otras vulnerabilidades ya conocidas por la forma en la que está hecha la aplicación en si.

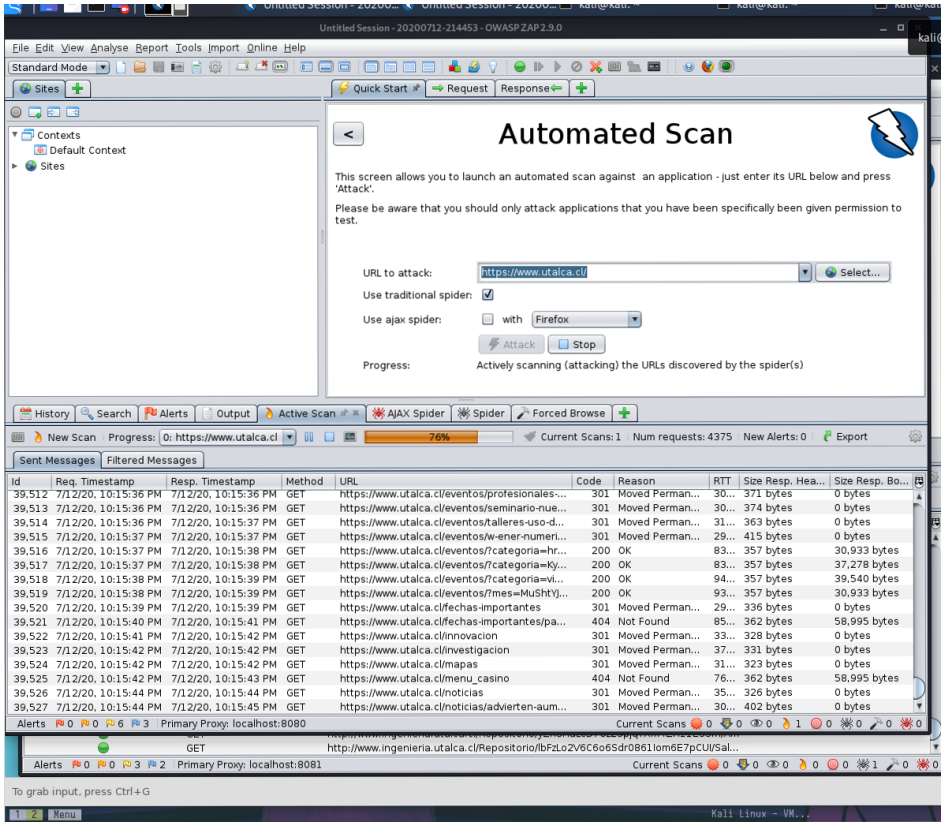
1.1. Actividad 1 - NMAP

```
kali@kali:~$ nmap -sV --version-intensity 5 www.utalca.cl
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-12 22:29 -04
Nmap scan report for www.utalca.cl (134.209.223.104)
Host is up (0.14s latency).
Not shown: 995 filtered ports
PORT     STATE  SERVICE     VERSION
22/tcp   open   ssh         OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp   open   http        nginx 1.15.8
443/tcp  open   ssl/http    nginx 1.15.8
8080/tcp closed http-proxy
9000/tcp open   cslistener?
1 service unrecognized despite returning data. If you know the service/version, please submit the follow
ing fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port9000-TCP:V=7.80%I=5%D=7/12%Time=5F0BC78A%P=x86_64-pc-linux-gnu%r(Ge
SF:tRequest,1630,"HTTP/1\.1\x20200\x20\r\nX-Frame-Options:\x20SAMEORIGIN\r
SF:\nX-XSS-Protection:\x201;\x20mode=block\r\nX-Content-Type-Options:\x20n
SF:osniff\r\nCache-Control:\x20no-cache,\x20no-store,\x20must-revalidate\r
SF:\nvary:\x20accept-encoding\r\nContent-Type:\x20text/html;charset=utf-8\
SF:r\nDate:\x20Mon,\x2013\x20Jul\x202020\x2002:31:38\x20GMT\r\nConnection:
```

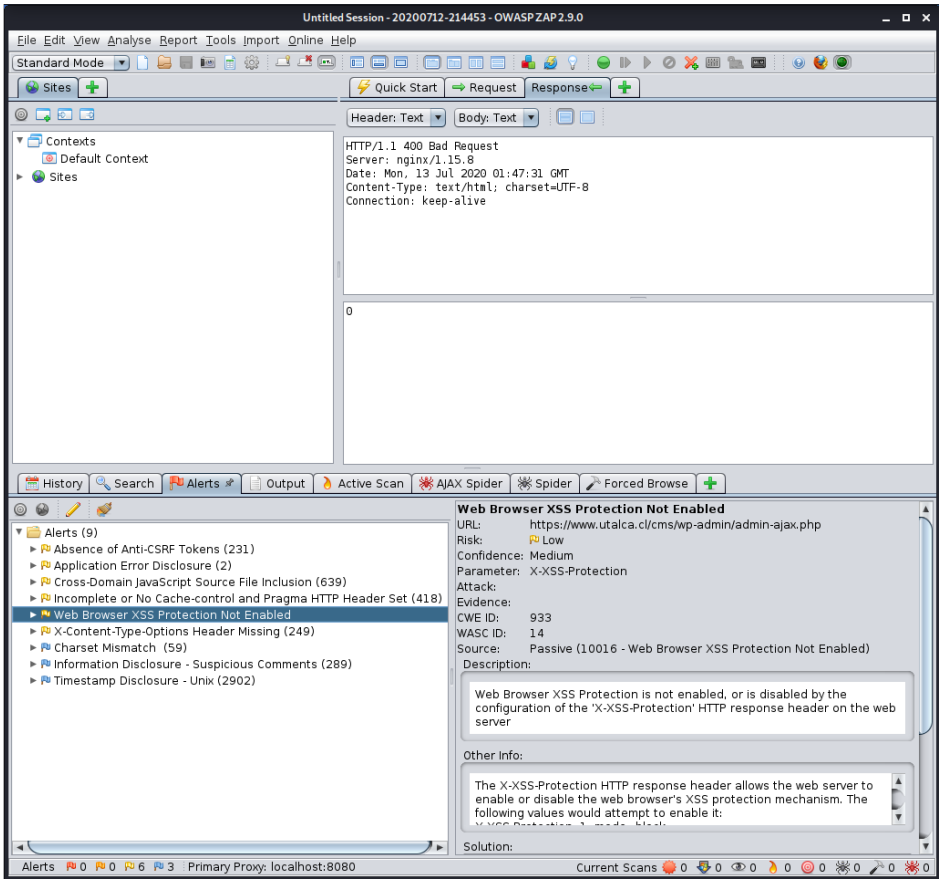
El primer análisis entregado por NMAP no es muy alentador. A diferencia de ocasiones anteriores, en este caso al parecer solo están abiertos los puertos para el consumo de tráfico web y un openssh que está protegido con un acceso solo por llave privada (bien jugado, aprendieron).

Adicionalmente hay un puerto abierto, el 9000 que típicamente en estos casos queda abierto para ser utilizado por sentry, lo cual no debería de ser tan sorprendente.

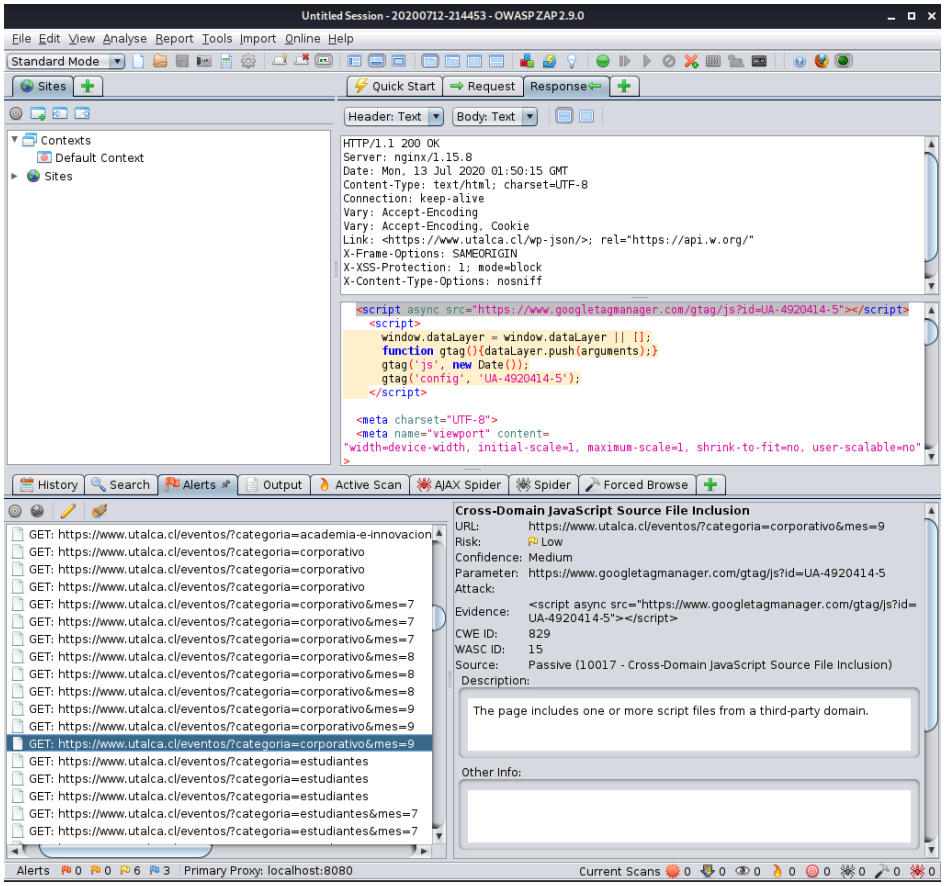
1.2. Actividad 2 - OWASP ZAP



Inicio de la ejecución de OWASP ZAP. Utilizamos los parámetros por defecto el cual ejecuta una búsqueda estandar. No hay necesidad de probar otras arañas ya que la diferencia por headers es nula. Durante esta ejecución es posible apreciar dos procesos principales. La araña que se encarga de buscar la información del sitio y realizar un mapa de este y la ejecución activa de los ataques -aka probing-.



Podemos ver una vez terminada la ejecución de ZAP que no se encuentran mayores vulnerabilidades, aunque llama la atención el número de alertas que son lanzadas por la herramienta en si.

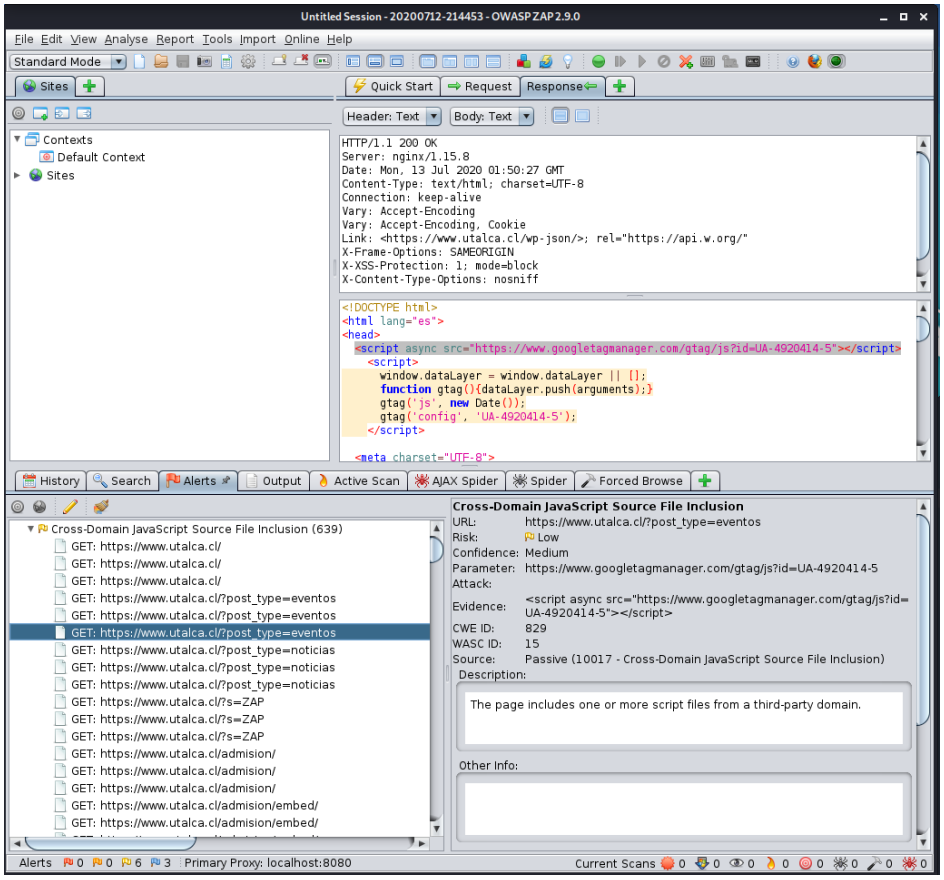


En si, la ejecución de ZAP como muchas herramientas automatizadas para la búsqueda de vulnerabilidades es muy propensa a encontrar falsos positivos. Por ejemplo, tomemos este caso de supuesta ejecución remota.

Podemos apreciar que se incluye un archivo desde google tag manager, por lo que si, es un problema el estar incluyendo archivos de una fuente de terceros. Sin embargo, el objetivo de este laboratorio es encontrar vulnerabilidades dentro de la aplicación en si y esta solo afecta al cliente, por ejemplo, cuando este dominio está pasando por un ataque MITM entonces el cliente puede ser afectado al cargar un script diferente.

Sin embargo, no es un problema propio de la aplicación. De todas maneras, esta información es valiosa al momento de realizar un assesment, ya que nos sugiere que un buen vector de ataque lateral son usuarios que acceden al sitio y poseean credenciales y con los que podamos tener contacto dentro de la misma red.

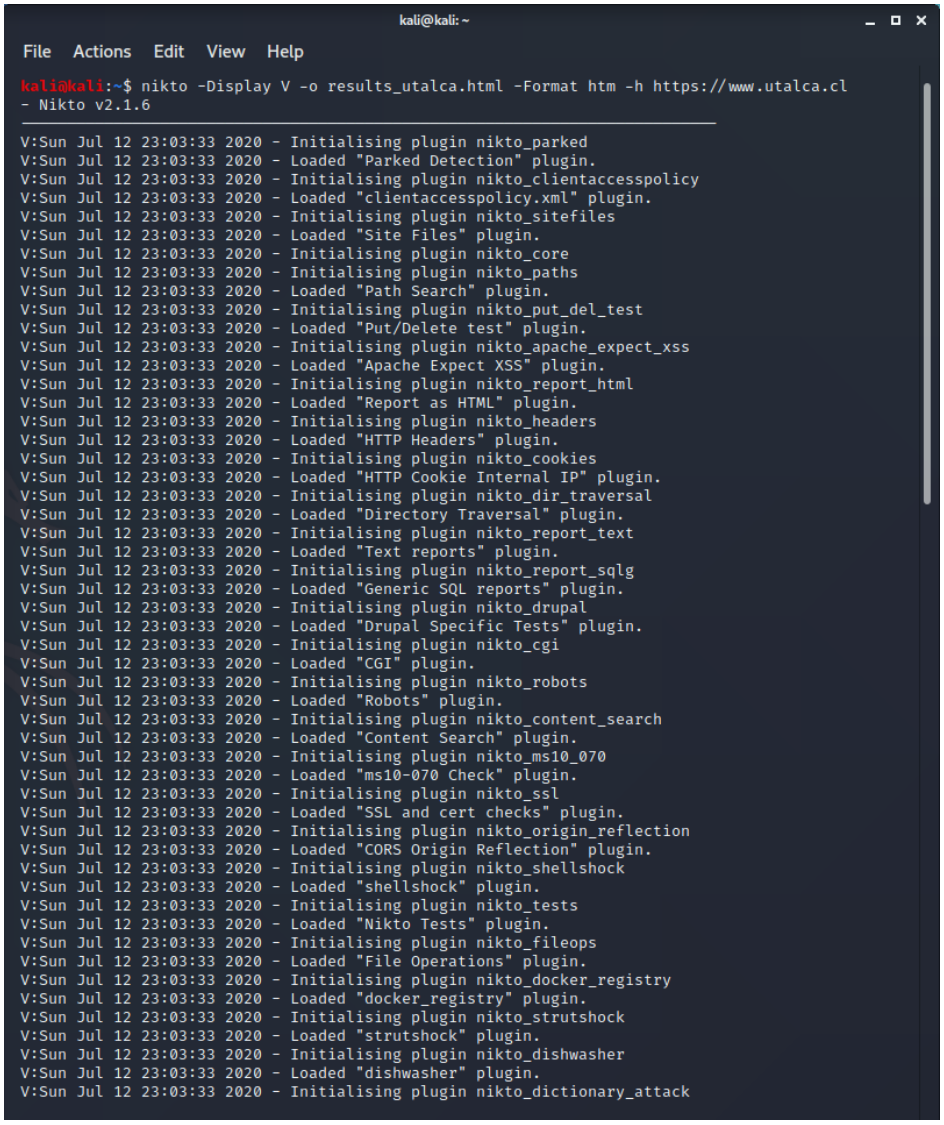
La existencia de estas URL con parámetros nos sugiere un buen punto de partida para probar inyecciones SQL.



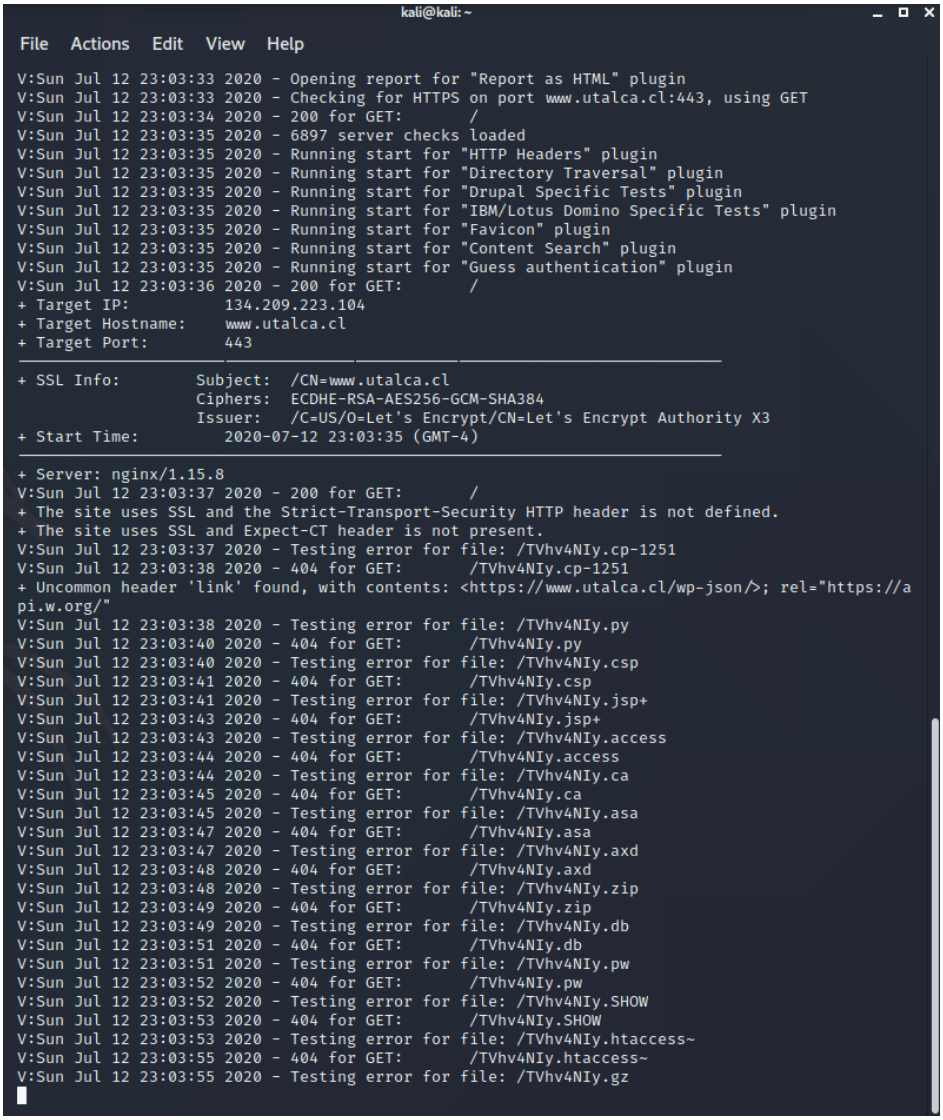
Sin embargo, al revisar mas en profundidad las alertas entregadas, además de confirmar los falsos positivos, encontramos que en realidad la razón de la gran cantidad de estos es porque derechamente hay una cantidad enorme de malas prácticas de programación involucradas en el desarrollo del sitio (sin contar que es un wordpress).

1.3. Actividad 3 - NIKTO

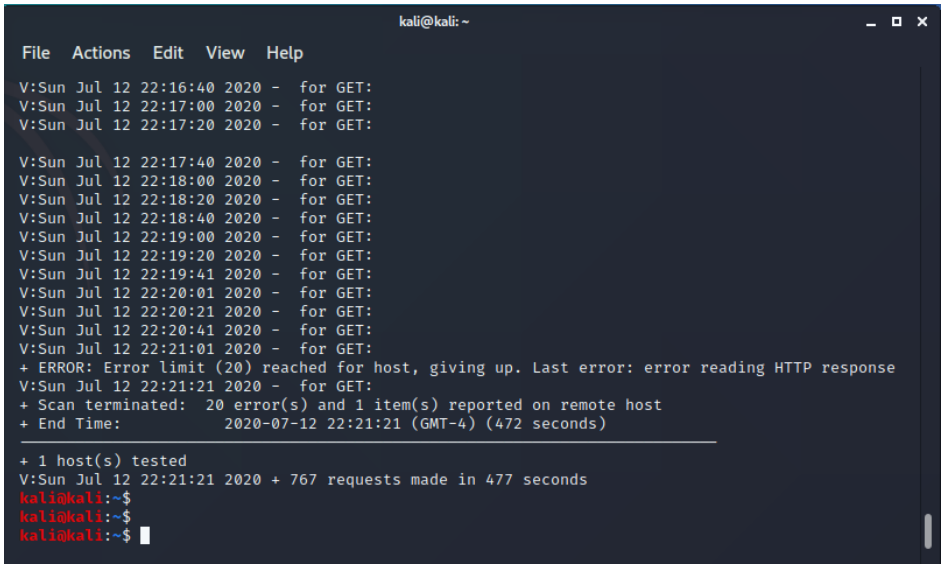
Para este caso utilizamos el siguiente comando para almacenar la información dentro de un archivo: `nikto -Display V -o results.html -Format htm -h https://www.utalca.cl`.



Sin embargo nos topamos con un problema fundamental, sin embargo que nos impide utilizar esta herramienta por completo. Al parecer el servidor tiene tiempos de respuesta relativamente altos para las consultas, no a su vez con la resolución de recursos estáticos.



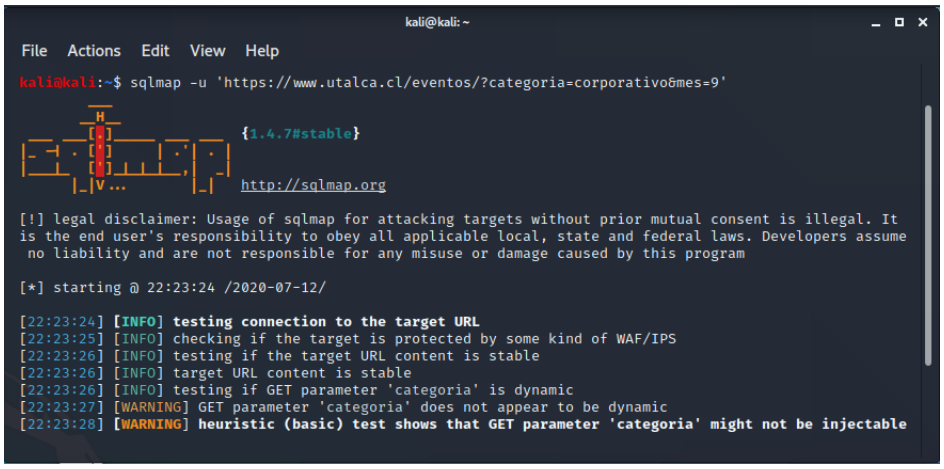
Debido a que NIKTO prueba los ataques de manera secuencial, al momento de ejecutar un ataque para poder revisar los archivos disponibles, tenemos un retraso de aproximadamente 500ms a 1600ms. Hay dos hipótesis al respecto, la primera es que toda esta plataforma está bajo un tarpit, la segunda es que el servidor está bajo mucho estrés o bien está mal configurado. Históricamente se ha probado que es mas probable que sea la segunda opción, sin embargo, este problema nos impide resolver el comando dentro de un tiempo relativamente razonable".



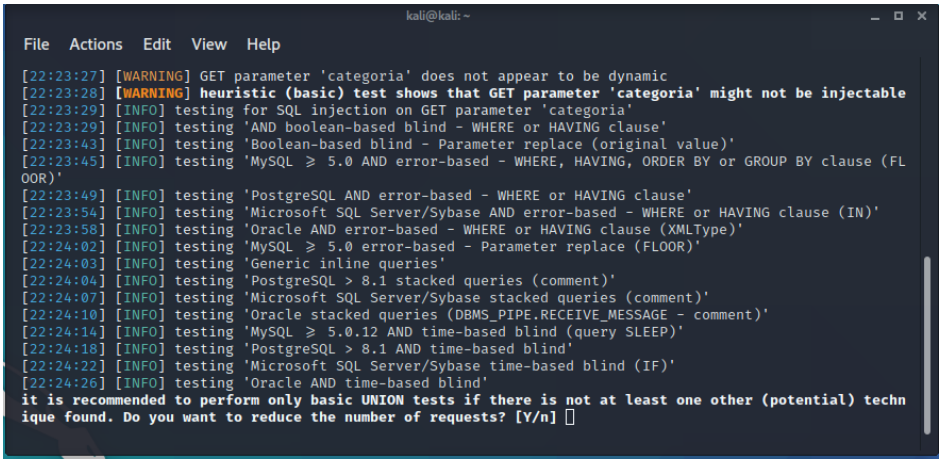
Ahora el problema de esto no es que sea lento el análisis (nikto por naturaleza es lento), si no que el analisis en vez de tomar horas, pasa a tomar días solo por el efecto del tarpit. Incluso paralelizando los análisis, no se obtienen resultados conluyentes ya que no alcanza a realizar todo el trabajo en un tiempo relativamente razonable (digamos, unas 5 horas).

1.4. Actividad 3 - SQLMAP

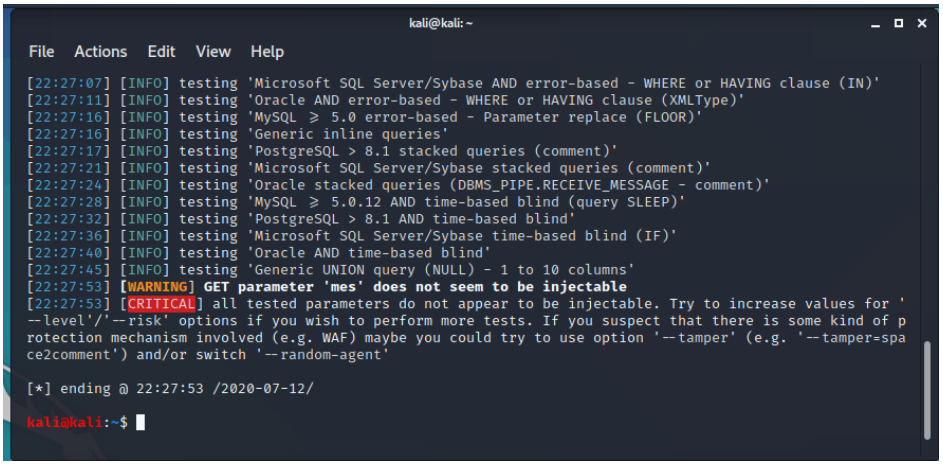
Para efectos de simpleza solo se muestra una ejecución.



Utilizando la url descubierta en el paso anterior (en realidad se probó con un conjunto de estas), procedemos a ejecutar SQLMAP sobre la dirección. Dado que no estamos necesariamente trabajando con scripts de inyección específicos, no utilizamos mayores argumentos.



Sin embargo en los primeros 10 minutos podemos ver que la ejecución se detiene dentro del modo interactivo indicando que no hay vectores de ataque por inyección por tanto sugiere cambiar la estrategia. A esto se le dice que si y continúa el análisis.



Sin embargo al cabo de un tiempo, notamos que no arroja mayores resultados. Esto pasó con todas las URL probadas que salieron de la ejecución de la araña de ZAP.

2. Conclusiones

Si bien todas las herramientas fallaron, esto no quiere decir que la aplicación esté libre de fallas. En primer lugar, todos las pruebas fueron ejecutadas desde una máquina en la red, por tanto la situación podría ser diferente al intentarlos dentro de la misma red de la universidad.

Adicionalmente, como se pudo observar durante la ejecución de ZAP, la aplicación en si cuenta con malas prácticas de programación por parte de los módulos de terceros invlucrados en esta.

De ejecutarse un análisis, este no solo debe hacerse a nivel de la aplicación en si como lo

especifica el enunciado, se tiene que tomar en cuenta otros sitios los cuales puedan servir como vectores laterales, sitios relacionados, revisar la estructura de la red y los servicios ya que en este caso, resulta que el sitio en cuestion es simplemente un front-end para otra aplicación conectada por detras.

Referencias

[1] Documentación de sqlmap *sqlmap homepage*. http://sqlmap.org/

[2] Documentación de nmap *nmap homepage*. https://nmap.org/

[3] Documentación de OWASP ZAP *OWASP ZAP project homepage*. https://owasp.org/www-project-zap/

[4] RedTeamTutorials *NIKTO Cheat Sheet*. https://redteamtutorials.com/2018/10/24/nikto-cheatsheet/