Detección de anomalías de tráfico en servidores web

Mitsiu Alejandro Carreño Sarabia
E23S-18014

1. Introducción

Se propone un método basado en modelos no supervisados de aprendizaje máquina para evaluar y detectar valores anómalos en las peticiones que recibe un servidor web. Mediante este método es posible evaluar nuevas peticiones basado en el tráfico histórico del servidor y obtener un índice de similitud respecto a solicitudes pasadas, con ello es posible detectar anomalías o contenido malicioso y tomar acciones correctivas.

2. Problemática

Con la expansión del acceso a servicios de internet, así como la creciente disponibilidad de dispositivos de distintas categorías para conectarse a la red, la demanda y tráfico de servicios web se encuentra en constante aumento. Mucho se ha desarrollado en términos de escalabilidad de infraestructura así como adopción de soluciones distribuidas para dar servicio a la creciente demanda.

La cantidad de tráfico que recibe un servidor web con acceso a internet es inmenso, el cuál recibe peticiones de usuarios reales, peticiones de bots, peticiones automatizadas y peticiones de usuarios con intenciones maliciosas, solo analizar la cantidad de información generada de manera manual es una tarea imposible.

El tráfico de servidores web tiene claras tendencias como recursos solicitados, región geográfica de donde se solicita, hora en que se solicitó, cantidad de bytes enviados, por lo que identificar las tendencias y detectar las anomalías es un trabajo que puede ser automatizado.

Dada la tendencia de alojar datos sensibles aumenta la importancia a evaluar qué y cómo se están accediendo a los recursos solicitados, así como desarrollar herramientas que faciliten filtrar las anomalías para tomar acciones correctivas.

3. Análisis de Topología

Regresión lineal: Dado que las horas de mayor demanda a los recursos del servidor son marcadas y denota una clara tendencia en horarios de trabajo y fines de semana, es posible realizar una regresión lineal para predecir las horas en las que se demandará el servidor.

K-means: Se decidió agrupar las peticiones partiendo de la cantidad de bytes enviados y el estatus HTTP de la respuesta, en donde se nota una clara tendencia a que las respuestas exitosas (2XX-3XX) contesten más información que una respuesta de error (4XX-5XX) y cabe destacar un tercer cluster de una respuesta que a pesar de ser exitosa, tuvo una respuesta con mucha más información que el resto de respuestas.

Complejos simpliciales: Para la elaboración de los complejos simpliciales se comienza por graficar cada punto (renglón) con todas sus dimensiones (características) y estos se toman como base para generar los 0-simplices, es decir una nube de puntos multidimensional.

De estos puntos se propone conectar en 1-simplices (bordes) según el atributo “domain” el cual comprende subdominio (si existe) dominio y dominio de nivel superior (por ejemplo “stage.puntoderecarga.mx”) de esta manera se tiene una serie de 1-simplices (líneas) que conecta las peticiones según el subdominio y dominio al que apuntan.

A fin de generar los 2-simplices (triángulos) se puede tomar el atributo “status” el cuál

hace referencia al código de estado HTTP que se respondió a cada petición[1] en este punto se tendrá un conjunto de triángulos donde por cada triángulo corresponde al mismo dominio y subdominio y estatus.

Finalmente para generar los 3-simplices (tetraedros) es posible agregar el atributo “clean_path” el cuál contiene los directorios y subdirectorios (si existen) para llegar al recurso solicitado (por ejemplo “/intra/acceso_i.php”) cabe destacar que el atributo “clean_path” descarta cualquier parámetro o query de la url por lo que únicamente tiene la ruta y el recurso solicitado, de esta manera se controla la variabilidad de los parámetros.

Con la topología descrita se puede realizar un análisis de relación entre el código de estado HTTP y el recurso solicitado desagregado por cada uno de los sitios alojados, con ello es sencillo detectar solicitudes a recursos anómalos y evaluar cómo manejo el servidor dicha solicitud, también es posible detectar si el servidor está manejando solicitudes esperadas con tendencia a responder errores y realizar las correcciones correspondientes.

1. https://es.wikipedia.org/wiki/Anexo:C%C3%B3digos_de_estado_HTTP