



INCIDENTES DE CIBERSEGURIDAD

Unidad 1. Actividad 34



02 DE ABRIL DE 2024
CARLOS DÍAZ MONTES
ESPECIALIZACIÓN DE CIBERSEGURIDAD

Índice

Enunciado.....	2
REALIZA UN INFORME SOBRE -- flatbuffers.....	2
¿Qué es la flatbuffers?	2
Características principales	2
Como se inicia	3
Funcionalidades de flatbuffers para snort	4
Conclusión	4

Enunciado

INDICA SU FUNCIONALIDAD PARA SNORT.-

REALIZA UN INFORME SOBRE-- flatbuffers.

¿Qué es la flatbuffers?

FlatBuffers es una biblioteca de serialización multiplataforma diseñada para lograr la máxima eficiencia de la memoria. Le permite acceder directamente a datos serializados sin analizarlos ni descomprimirlos primero, sin dejar de tener una excelente compatibilidad hacia adelante y hacia atrás.

Características principales

Las características principales son:

- **Estructura de datos sin procesar:** FlatBuffers almacena los datos en un formato binario directamente accesible sin necesidad de un paso de descompresión o análisis. Esto elimina la necesidad de copiar datos y permite un acceso directo a los mismos.
- **No requiere un esquema externo:** A diferencia de algunas otras bibliotecas de serialización, FlatBuffers no requiere un archivo de definición de esquema externo. En su lugar, utiliza un lenguaje de esquema integrado (IDL, por sus siglas en inglés) para definir la estructura de datos en el código fuente.
- **Serialización y deserialización eficientes:** FlatBuffers está diseñado para minimizar el tiempo y la memoria necesarios para serializar y deserializar datos. Esto lo hace ideal para aplicaciones donde el rendimiento es crítico, como juegos y aplicaciones de realidad virtual.
- **Soporte multiplataforma:** FlatBuffers es compatible con una amplia variedad de plataformas, incluyendo Windows, macOS, Linux, Android, iOS y más. Esto lo hace adecuado para aplicaciones que necesitan interoperabilidad entre diferentes sistemas.
- **Flexibilidad:** FlatBuffers ofrece flexibilidad en la manipulación de datos, permitiendo la modificación de los datos serializados sin necesidad de deserializarlos por completo. Esto es útil en casos donde solo se necesita acceder a una parte de los datos o actualizar ciertas partes de ellos.
- **Soporte para datos en streaming:** FlatBuffers puede manejar datos en streaming, lo que significa que los datos pueden ser procesados mientras se están leyendo o escribiendo, en lugar de esperar a que se completen.
- **Sistemas Operativos compatibles:** Los sistemas compatibles con Flatbuffers son Windows, macOS, Linux, Android y Y cualquier otro con un compilador de C++ reciente (C++ 11 y posterior).

Como se inicia

1. Construya el compilador para flatbuffers (flatc)

Úselo cmakepara crear los archivos de compilación para su plataforma y luego realice la compilación (ejemplo de Linux).

```
cmake -G "Unix Makefiles"
make -j
```

2. Defina su esquema de búfer plano (.fbs)

Escriba el esquema para definir los datos que desea serializar. Por ejemplo:

```
1 // Example IDL file for our monster's schema.
2
3 namespace MyGame.Sample;
4
5 enum Color:byte { Red = 0, Green, Blue = 2 };
6
7 union Equipment { Weapon } // Optionally add more tables.
8
9 struct Vec3 {
10   x:float;
11   y:float;
12   z:float;
13 }
14
15 table Monster {
16   pos:Vec3;
17   mana:short = 150;
18   hp:short = 100;
19   name:string;
20   friendly:bool = false (deprecated);
21   inventory:[ubyte];
22   color:Color = Blue;
23   weapons:[Weapon];
24   equipped:Equipment;
25   path:[Vec3];
26 }
27
28 table Weapon {
29   name:string;
30   damage:short;
31 }
32
33 root_type Monster;
```

3. Genera código para tu(s) idioma(s)

Utilice el flatccompilador para tomar su esquema y generar código específico del idioma:

```
./flatc --cpp --rust monster.fbs
```

Que genera monster_generated.hy monster_generated.rsarchivos.

4. Serializar datos

Utilice el código generado, así como el FlatBufferBuilderpara construir su búfer serializado. (C++ejemplo)

5. Transmitir/almacenar/guardar buffer

Utilice su búfer serializado como desee. Envíalo a alguien, guárdalo para más tarde, etc...

6. leer los datos

Utilice los descriptores de acceso generados para leer los datos del búfer serializado.

No es necesario que sea el mismo idioma/versión de esquema, FlatBuffers garantiza que los datos sean legibles en todos los idiomas y versiones de esquema.

Funcionalidades de flatbuffers para snort

Puede ser utilizado en el contexto de Snort de varias maneras:

- **Serialización eficiente de datos:** FlatBuffers puede ayudar a serializar y deserializar datos de manera eficiente, lo que puede ser útil para procesar y almacenar reglas de detección de intrusiones, registros, eventos, y otros datos utilizados en Snort.
- **Interoperabilidad entre plataformas:** FlatBuffers puede utilizarse para facilitar la comunicación entre diferentes componentes de un sistema de detección de intrusiones, permitiendo el intercambio eficiente de datos entre ellos en diferentes plataformas.
- **Manipulación eficiente de datos en memoria:** La capacidad de FlatBuffers para acceder a datos serializados en memoria de manera eficiente puede ser útil para realizar operaciones de filtrado, análisis y procesamiento de paquetes de red en Snort.
- **Soporte multiplataforma:** FlatBuffers es compatible con una amplia gama de plataformas, lo que permite a Snort ejecutarse en diferentes sistemas operativos y arquitecturas de hardware.

Conclusión

En conclusión, FlatBuffers es una potente biblioteca de serialización de datos diseñada para ofrecer un rendimiento eficiente y un uso óptimo de la memoria. Sus características principales incluyen la capacidad de almacenar datos en un formato binario directamente accesible, la flexibilidad para definir estructuras de datos sin un esquema externo, la eficiencia en la serialización y deserialización de datos, y el soporte multiplataforma. Estas características hacen que FlatBuffers sea ideal para una amplia gama de aplicaciones, desde sistemas de detección de intrusiones hasta juegos y aplicaciones de realidad virtual, donde el rendimiento y la eficiencia son críticos. Con su enfoque en la optimización de recursos y su facilidad de uso, FlatBuffers se ha convertido en una herramienta invaluable para desarrolladores que buscan maximizar el rendimiento de sus aplicaciones mientras minimizan el consumo de recursos.