

# Open Source Software Team Mission 1

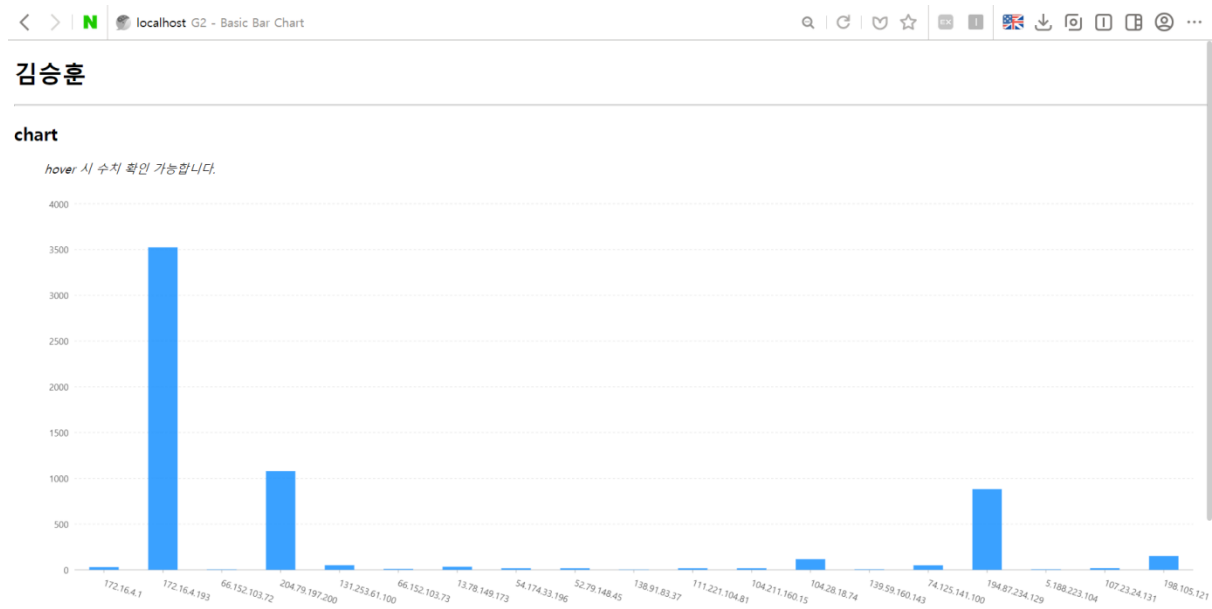
## Experiencing D3.js

Group #7 김승훈

조장 배수범

조원 이용혁 이상무 이진하 최승도 유지현

Step 1 수업시간에 진행한 데이터 시각화를 따라해보고 결과화면을 팀이름이 나오도록 캡처하여 제출하시오.



데이터셋은 공통적으로 <http://malware-traffic-analysis.net/training-exercises.html> 에 있는

2017-01-28 – Traffic analysis exercise – Thanks, Brian. 을 사용하였다.

첫번째 데이터 시각화는 d3js.org 에 있는 basic-bar-chart를 사용하였다. 기본적인 그래프 형태인 막대그래프 형태이고 x축은 source IP, y축에는 해당 source IP의 패킷 수를 보여주는 방식으로 가공하였다. 해당 방식을 채택한 이유는 간편하게 해당 데이터셋의 각각의 source IP의 양을 시각화하여 자주 사용되는 IP를 한눈에 알아볼 수 있고 사용빈도를 명확하게 구별할 수 있기 때문이다.

두번째 데이터 시각화는 d3js.org에 있는 chord diagram를 사용하였다. 원형 형태로 선의 넓이(범위)는 두 IP간의 통신에서 사용한 packet 수 , 색은 이다. 해당 방식을 채택한 이유는 source IP와 destination IP를 연결함으로써 어떤 IP들 간에 통신빈도를 명확하게 알 수 있게 하기위함이다.

3) D3.js를 이용하여 주어진 보안 dataset를 시각화한 결과 화면과 소스코드를 압축하여 제출하시오.

■ Bar chart 데이터 시각화

● 소스코드에 대한 간단한 설명

```
name: "data",
value: (function(){return(
[
  {
    year: '1951',
    sales: 38
  },
  {
    year: '1952',
    sales: 52
  },
  {
    year: '1956',
    sales: 61
  },
  {
    year: '1957',
    sales: 145
  },
  {
    year: '1958',
    sales: 48
  },
  {
    year: '1959',
    sales: 38
  },
  {
    year: '1960',
    sales: 38
  },
  {
    year: '1962',
    sales: 38
  }
]
}))
```

원본 그래프 데이터(js)의 모습으로 본 조의 데이터셋을 넣기위해 코드를 추가/수정해야 한다.

```

1 var packet = [{"No":1,"Time":0,"Source":"172.16.4.1","Desti
2 {"No":2,"Time":0.008974,"Source":"172.16.4.193","Destinatio
3 {"No":3,"Time":0.107795,"Source":"172.16.4.193","Destinatio
4 {"No":4,"Time":1.52807,"Source":"172.16.4.193","Destinatio
5 {"No":5,"Time":1.636618,"Source":"172.16.4.193","Destinatio
6 {"No":6,"Time":2.060191,"Source":"172.16.4.193","Destinatio
7 {"No":7,"Time":2.060388,"Source":"172.16.4.193","Destinatio
8 {"No":8,"Time":2.060396,"Source":"172.16.4.193","Destinatio
9 {"No":9,"Time":2.136179,"Source":"172.16.4.193","Destinatio
10 {"No":10,"Time":2.136377,"Source":"172.16.4.193","Destinati
11 {"No":11,"Time":2.136383,"Source":"172.16.4.193","Destinati
12 {"No":12,"Time":2.603784,"Source":"172.16.4.193","Destinati
13 {"No":13,"Time":2.900156,"Source":"172.16.4.193","Destinati

```

데이터셋을 패킷으로 묶음

```

1 // json 형식의 변수를 불러옵니다.
2 import packet from './testPacket.js';
3
4 // 중복을 제거하는 기능입니다.
5 function removeDuplicates(originalArray, prop) {
6   var newArray = [];
7   var lookupObject = {};
8
9   for(var i in originalArray) {
10     lookupObject[originalArray[i][prop]] = originalArray[i];
11   }
12
13   for(i in lookupObject) {
14     newArray.push(lookupObject[i]);
15   }
16   return newArray;
17 }
18
19 // 불러온 데이터를 Source 라는 키 기준으로 중복을 제거합니다.
20 // 중복이 제거된 배열 uniqueArray
21 var uniqueArray = removeDuplicates(packet, "Source");
22
23 // 가공할 데이터를 담은 빈 배열을 만들어 둡니다.
24 let arr = [];
25
26 uniqueArray.map((d,i) => {
27
28   // 불러온 데이터중에 uniqueArray 의 i 번째 객체중 source가 일치하는
29   // 객체들을 골라 packets 에 담습니다.
30   const packets = _filter(packet, (p) => p.Source == d.Source)
31
32   // 새로운 객체를 생성합니다.
33   // 키는 source 와 packets
34   // 밸류는 소스 아이피와 그 아이피를 소스로 가진 패킷의 개수.
35   const obj = {Source: packets[0].Source, Packets: packets.length};
36
37   // 만들어진 객체를 만들어둔 빈 배열에 담습니다.
38   arr.push(obj);
39 });
40
41 // 가공한 데이터를 담아줍니다.
42 {
43   name: "data",
44   value: (function(){return(
45     arr
46   )})

```

데이터를 수정/가공 후의 모습(주석으로 코드설명)

- 텍스트를 통해 볼 때 보다 눈에 띄는 특징/의미

중복된 내용이 많아서 텍스트로 하나하나 보기에는 찾아가면서 읽어야 한다는 것이 불편했지만 시각화 했을 때는 정보가 한눈에 보이기 때문에 원하는 정보를 직관적으로 이해할 수 있기에 편하다.

거기에 오픈소스를 이용해 추가 기능을 이용했을 때, 힘들게 직접 구현하지 않아도 훨씬 깔끔하고 좀 더 눈에 띄는 효과를 얻을 수 있어서 굉장히 편리한 것 같다.

- Chord 데이터 시각화
- 소스코드에 대한 간단한 설명

```
name: "data",
inputs: ["d3"],
value: (async function(d3) {
  const imports = await d3.json("https://gist.githubusercontent.com" +
    "/mbostock/1044242" +
    "/raw/3ebc0fde3887e288b4a9979dad446eb434c54d08" +
    "/flare.json");
```

원본 그래프 데이터(js)의 모습으로 본 조의 데이터셋을 넣기위해 코드를 추가/수정해야 한다.

[illegible]

데이터셋을 해당 데이터 그래프에 맞춰 가공하였음  
(source(name),destination(imports),size)

```
{
  name: "data",
  inputs: ["d3"],
  value: (async function(d3) {
    const imports = await d3.json("./json.json");
```

데이터를 수정/가공 후의 모습(주석으로 간단한코드설명 소스코드에 포함하였음)

- 텍스트를 통해 볼 때 보다 눈에 띄는 특징/의미

첫번째 시각화 데이터와 마찬가지로 텍스트로 IP 의 사용빈도를 찾는 것은 불편하나 시각화를 통해 IP 간의 통신빈도가 한 눈에 보이기 때문에 직관적인

view 를 제공할 수 있다. 이는 데이터를 읽는 것에 대한 편리성, 가독성에 의의가 있다.

github link : [https://github.com/BaeSooBeum/OSS\\_TeamKSH](https://github.com/BaeSooBeum/OSS_TeamKSH)