Snort 를 이용한 PSD(Payload Size Distribution)기반 응용 트래픽 분류

김성민, 유종현, 이수강, 김명섭

고려대학교 컴퓨터정보학과 네트워크관리연구실

{gogumiking, dididaboa, sukanglee, tmskim}@korea.ac.kr

요 약

오늘날 인터넷을 사용하는 다양한 응용프로그램들이 등장하고 인터넷의 보편화로 인해 트래픽이 복잡 다양해지고 있다. 이러한 상황 속에서 네트워크의 효과적인 운용과 관리를 위해 네트워크 트래픽 분석의 중요성은 나날이 증가하고 있다. 통계정보를 이용하여 트래픽을 분석하는 방법 중 하나인 PSD(Payload Size Distribution) 기반 트래픽 분류 방법은 플로우의 첫 N 개의 패킷의 특성을 해당 응용의 시그니처로 만들어 트래픽을 분류하는 방법이다. 따라서 본 논문에서는 PSD 기반 트래픽 분류 방법에서 사용되는 응용 별 시그니처를 오픈소스 패킷 탐지 도구 중 하나인 Snort 를 이용하여 구현할 수 있다는 가능성을 검증하고, 성능을 평가하는데 목적이 있다.

1. 서론

오늘날 초고속 인터넷 보급과 급격한 네트워크 장비의 발달로 인해 발생하는 트래픽이 복잡 다양해지고 있다. 이러한 상황 속에서 네트워크의 효과적인 운용과 관리를 위해서는 네트워크 트래픽 분석은 필수적인 요소이다.

트래픽을 분류하는 여러 방법[1] 중 하나인 통계정보를 이용한 분류 방법은 패킷의 크기, 전송 방향, 수집 시간, Inter-arrival time 등을 feature 로 사용한다. 그 중 PSD(Payload Size Distribution) 기반 트래픽 분류 방법[2]은 플로우의 처음 N 개 패킷들의 페이로드 길이와 방향을 사용하여 해당 응용을 탐지한다.

본 논문에서는 PSD 기반 트래픽 분류 방법에서 사용되는 시그니처를 패킷 분석 도구 중 하나인 Snort[3]의 규칙(rule)을 이용하여 구현할 수 있다는 가능성을 제안하고, 성능을 검증하고자 한다.

본 논문은 다음과 같은 순서로 기술한다. 2 장에서는 기존 PSD 기반 트래픽 분류 방법과 PSD 기반 시그니처 요소들의 기능을 수행하는 Snort 규칙의 기능을 소개한다. 3 장에서는 기존 시그니처와 Snort 를 이용하여 동일 트래픽을 분석하여 비교 성능평가를 하고, 마지막으로 4 장에서는 결론 및 보완점, 향후 연구에 대하여 기술한다.

이 논문은 2012 년 정부(교육과학기술부)의 재원으로 한국연구재단(2012R1A1A2007483) 및 2013 년도 정부(미래창조과학부)의 재원으로 한국연구재단-차세대정보.컴퓨팅기술개발사업(2010-0020728)의 지원을 받아 수행된 연구임.

2. 본론

본 장에서는 통계 정보를 이용한 응용 분석 방법 중에 하나인 PSD 기반 응용 분류 방법과 PSD 기반 시그니처 요소들의 기능을 수행하는 Snort 규칙의 기능에 대하여 간략히 소개한다.

2.1 PSD 기반 트래픽 분류 방법

통계 정보를 이용한 응용 분석 방법은 수집된 패킷의 크기, 윈도우 크기, 그리고 수집 시간 등을 feature 로 하여 해당 패킷을 발생시킨 응용을 탐지, 분류하는 방법이다. 그 중 PSD 기반 응용 분류 방법은 페이로드의 크기 분포를 벡터형태의 시그니처로 사용하여 트래픽을 분류하는 방법이다.

표 1. PSD 기반 시그니처 예

① sig id	27
② protocol	TCP
③ port	5004
④ payload size[final]	[28, -1460, -1460, -1460, -1460]
⑤ packet threshold	[13, 0, 0, 0, 0]

표 1 은 PSD 기반 시그니처의 한 예로 ①은 시그니처의 고유 식별번호 이며, ②와 ③은 protocol 의 종류와 port 번호를 나타낸다. ④는 순서대로 수집된 각 패킷의 크기를 나타내고, 부호를 통해 각 패킷의 방향을 표현한다. 여기서 '+'는 서버로 향하는 패킷, '-'는 클라이언트로 향하는 패킷이다. 마지막으로 ⑤는 각 패킷크기의 임계값을 나타낸다.

PSD 기반 트래픽 분류 방법은 새롭게 수집되는 트래픽 플로우의 처음 N 개의 패킷과 표 1 에서 정의한 것과 같은 PSD 시그니처를 비교하여 특정 응용을 탐지한다.

2.2 PSD 기반 시그니처와 Snort 규칙

표 2. PSD 기반 시그니처와 Snort Rule Options

PSD Signature	Snort Rule Options
sig id	sid
protocol	[protocol]
port	[port]
payload size[final]	dsize, flow, flowbits
packet threshold	

표 2 는 기존의 시그니처와 이를 Snort 규칙에서 대체 할 수 있는 옵션들이다. 기존 PSD 시그니처의 'sig id'는 Snort 에서 'sid'라는 옵션으로 대체 가능하며, Snort 규칙 헤더부분에 protocol 종류와 port 번호를 적용한다. 또한 'payload size[final]'의 페이로드 크기정보와 'payload threshold'정보를 이용하여 Snort 에서는 'dsize'옵션으로 분석할 패킷 벡터의 크기를 정할 수 있다. 또한 'flow'옵션으로 패킷의 벡터방향을 정할 수 있다. 마지막으로 'flowbits'옵션으로는 패킷이 발생하는 순서대로 규칙을 매칭 할 수 있다.

표 3. 변환한 Snort 규칙의 예

①	pass tcp any any -> any 5004 (sid: 1000001; flow: to_server; flowbits: set,N1_1; dsize: 14<>42;)
②	pass tcp any 5004 -> any any (sid: 1000002; flow: to_client; flowbits: isset,N1_1; flowbits: unset,N1_1; flowbits: set,N1_2; dsize: 1460;)
③	pass tcp any 5004 -> any any (sid: 1000003; flow: to_client; flowbits: isset,N1_2; flowbits: unset,N1_2; flowbits: set,N1_3; dsize: 1460;)
④	pass tcp any 5004 -> any any (sid: 1000004; flow: to_client; flowbits: isset,N1_3; flowbits: unset,N1_3; flowbits: set,N1_4; dsize: 1460;)
⑤	alert tcp any 5004 -> any any (msg: "nateon"; sid: 1000005; flow: to_client; flowbits: isset,N1_4; flowbits: unset,N1_4; dsize: 1460;)

표 3 은 표 1 에서 보여준 PSD 기반 시그니처의 예를 Snort 규칙의 형태로 변환 한 것을 나타낸 표이다. 각각의 규칙은 패킷 단위로 매칭되지만 ①번에서 ⑤번까지 규칙은 flowbits 옵션을 통해 1 개의 플로우 내에서 순차적으로 패킷을 분석할 수 있다. 순차적으로 매칭을 수행하고 마지막 ⑤번 규칙이 매칭되면 해당 플로우에 대해 미리 정의된 alert 으로 기록한다.

3. 실험 및 결과

본 장에서는 기존 PSD 시그니처와 Snort 로 작성한 시그니처의 성능 비교를 위해 동일한 트래픽을 분석하는 실험을 진행 하였다. 실험에서 사용된 응용은 터미널프로그램인 Xshell 과 PuTTY 를 선정하였다. Xshell 은 총 36,145 개의 패킷으로 이루어진 57 개의 플로우를 분석 하였으며, PuTTY 는 17,808 개의 패킷으로 구성된 71 개의 플로우를 분석 하였다. 또한 기존 PSD 기반 시그니처와 달리 Snort 는 트래픽을 패킷 단위로 분석을 한다. 따라서 Snort 는 5 개의 규칙이 순차적으로 모두 매칭되었을 때 마지막 패킷만 alert 으로 기록되므로 이 경우 해당 응용을 분류한 것으로 가정하였다.

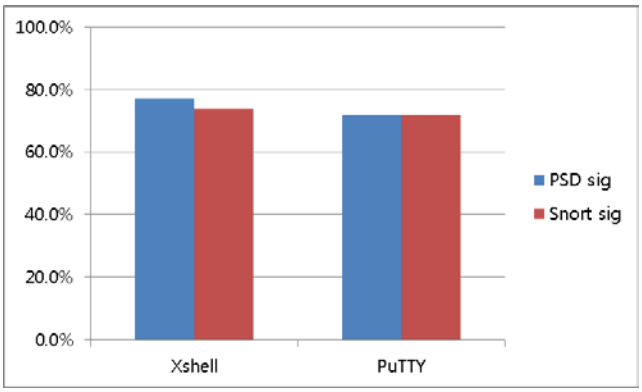

그림 1. 각 응용 별 분석률 비교

그림 1 은 기존 시그니처와 Snort 규칙으로 해당 응용을 탐지한 결과이다. 기존 PSD 기반 분석률과 Snort 규칙을 이용한 분석률이 거의 일치 하므로 Snort 를 PSD 기반 응용 트래픽 분류에 사용 가능하다. 그러나 Snort 를 이용한 분석기는 플로우 단위로 기록을 하지 못하고, 비연속적인 패킷임에도 규칙에 매칭이 되면 기록을 하는데, 이는 패킷 단위로 분석을 하는 Snort 의 기본적인 규칙의 한계를 보여준다. 결론적으로 Snort 를 사용하여 보다 정확한 트래픽 분석을 하기 위해서는 Snort 의 사용자 정의 옵션을 사용하여 고급기능을 구현해야 한다.

4. 결론

본 논문에서는 Snort 를 이용하여 PSD 기반 응용 분류 방법 구현의 가능성을 검증하고자 하였으며 실제로 가능성을 확인하였다. 그러나 Snort 는 트래픽을 플로우 단위가 아닌 패킷 단위로 분석을 하기 때문에 Snort 의 기본적인 규칙 옵션만으로는 기존 PSD 기반 트래픽 분류 방법보다 정확도가 떨어진다. 따라서 Snort 를 사용하여 보다 정확한 분석을 하기 위해서는 별도의 기능을 구현하여 고급기능을 사용해야 한다.

향후 연구로는 Snort 의 사용자 정의 옵션을 별도로 구현하여 Snort 를 이용한 PSD 기반 응용 트래픽 분류 방법의 분석률을 개선 시키고자 한다.

5. 참고 문헌

[1] 윤성호, 안현민, 김명섭, "다각적이고 계층적인 트래픽 분석을 위한 트래픽 분류 체계에 관한 연구 ", 정보처리학회 논문지 컴퓨터 및 통신시스템 제 3 권 제 2 호, Feb. 2014, pp. 47-56.

[2] 박진완, 윤성호, 박준상, 이상우, 김명섭, "통계시그니쳐 기반의 응용 트래픽 분류", 통신학회논문지 Vol.34 No.11, Nov., 2009, pp.1234-1244.

[3] http:// http://www.snort.org/snort