http://dx.doi.org/10.7840/kics.2015.40.7.1

시그니쳐 매칭 유형 분류를 통한 트래픽 분석 시스템의 처리 속도 향상

정 우 석•, 박 준 상°, 김 명 섭°

Performance Improvement of Traffic Identification by Categorizing the Signature Matching Type

Woo-Suk Jung•, Jun-sang Park°, Myung-Sup Kim°

요 약

응용 레벨 트래픽 분석은 네트워크의 효율적인 운영과 안정적인 서비스 제공을 위한 필수적인 요소이다. 응용 레벨 트래픽 분석을 위한 다양한 방법이 존재하지만 분류의 정확성, 분석률, 실용성을 고려했을 때 페이로드 시그니쳐 기반 분석 방법이 가장 높은 성능을 보인다. 하지만 페이로드 시그니쳐 기반 분석 방법은 다른 방법론에 비해 처리속도가 느리다는 단점이 있다. 본 논문에서는 각 시그니쳐가 페이로드에 매칭 되는 범위와 패킷의 순서 그리고 방향성과 같은 Offset value을 자동으로 추출하고 활용하여 시그니쳐를 매칭 유형별로 분류한다. 유형별로 분류된 시그니쳐에 최적화된 탐색범위를 지정하여 탐색범위를 최적화함으로써 페이로드 시그니쳐 기반 분석 방법의 처리 속도를 향상 시키는 방법을 제안한다.

Key Words : traffic analysis, signature matching type, payload signature, processing speed

ABSTRACT

The traffic identification is a preliminary and essential step for stable network service provision and efficient network resource management. While a number of identification methods have been introduced in literature, the payload signature-based identification method shows the highest performance in terms of accuracy, completeness, and practicality. However, the payload signature-based method's processing speed is much slower than other identification method such as header-based and statistical methods. In this paper, we first classifies signatures by matching type based on range, order, and direction of packet in a flow which was automatically extracted. By using this classification, we suggest a novel method to improve processing speed of payload signature-based identification by reducing searching space.

Ⅰ. 서 론

네트워크의 고속화와 더불어 다양한 서비스와 응용 프로그램이 개발됨에 따라 기업이나 개인들의 인터넷으로 대표되는 네트워크에 대한 의존이 상당히 커져가고 있다. 이러한 상황에서 네트워크의 효율적 운용

※ 본 연구는 BK21플러스 사업 및 2012년 정부(교육과학기술부) 의 재원으로 한국연구재단(2012R1A1A2007483)의 지원을 받아 수행된 결과임.
• First Author : Dept. of Computer and Information Science, Korea University. hary5832@korea.ac.kr, 학생회원
° Corresponding Author : Dept. of Computer and Information Science, Korea University. junsang_park@korea.ac.kr, 학생회원
° Corresponding Author : Dept. of Computer and Information Science, Korea University. tmskim@korea.ac.kr, 정회원
논문번호 : KICS2015-04-134, Received April 7, 2015; Revised June 9, 2015; Accepted June 9, 2015

과 관리를 위한 응용 레벨의 트래픽의 모니터링과 분석은 네트워크의 관리와 사용현황 파악, 그리고 확장 계획 수립 등의 다양한 분야에서 필요성이 커져가고 있다. 이를 위해서는 다양한 종류의 응용 레벨 트래픽을 정확하게 그리고 빠른 시간 안에 분류할 수 있는 방법이 요구된다.

응용 레벨 트래픽 분류 방법에 있어 페이로드 시그니쳐 기반 분석 방법은 패킷의 헤더 정보나 통계 정보 기반의 다른 분석 방법들에 비해 상대적으로 높은 분류 정확성과 분석률을 보인다[1,2]. 하지만 분류 시스템의 처리 속도에 있어 현재의 고속 네트워크상에서 발생하는 대용량 트래픽을 실시간으로 처리하기 위해서는 보다 빠른 방법이 요구 된다. 대용량의 트래픽을 발생시키는 응용 개수의 증가를 고려했을 때, 페이로드 기반 분석 방법의 처리 속도 향상은 해결해야 할 중요한 과제이다. 이를 해결하기 위해 기존의 다양한 연구에서는 패턴 매칭 알고리즘의 성능 개선 기법에 대한 연구가 주로 행해졌다[3-6]. 하지만 매칭 알고리즘의 성능 개선은 제한적이다. 또한 알고리즘별로 차이는 있지만, 일반적으로 시그니쳐와 페이로드를 매칭하는 과정에서 전체 페이로드를 대상으로 검사하게 된다. 페이로드 시그니쳐의 개수가 증가 할수록 응용 트래픽 분석률은 증가하지만, 각각의 시그니쳐가 일반적으로 매칭 되는 위치 값을 벗어난 영역을 탐색하는 횟수가 증가하여 불필요한 탐색이 증가하고 이로 인해 처리 속도가 감소하게 된다. 이러한 문제 해결을 위해 snort와 같은 침입 탐지 시스템은 Offset rule을 제공함으로써 탐색 범위를 제한할 수 있도록 한다. 하지만 snort rule에서 Offset rule 사용은 필수가 아닌 선택이므로 사용자에 따라 생성된 시그니쳐가 달라진다. 또한, Offset rule을 적용하기 위해 필요한 시그니쳐의 Offset을 추출하는 과정을 수동으로 할 경우 정확성이 떨어지는 문제와 시간이 오래 걸리는 문제가 발생한다. 본 논문에서는 시그니쳐에서 Offset을 추출하는 과정을 자동화하여 보다 정확하고 빠르게 다양한 Offset을 추출하고, 추출한 Offset value들을 활용하여 시그니쳐를 매칭 유형별로 분류한다. 매칭 유형별로 분류된 각각의 시그니쳐 별로 최적화된 탐색 범위를 지정하기 때문에 snort보다 효율적인 탐색 범위 축소가 가능하다.

본 논문의 구성은 다음과 같다. 본 장의 서론에 이어 2장에서는 관련연구에 대해 기술하고, 3장에서는 제안하는 방법의 배경이 되는 기존의 시그니쳐 매칭 방법의 문제점을 대해 기술한다. 4장에서는 시그니쳐의 매칭 유형을 고려한 트래픽 분석 방법에 대해 기술한다. 5장에서는 제안하는 방법을 트래픽 분석 시스템에 적용하고 성능 평가를 통해 그 효율성을 증명한다. 마지막으로 6장에서는 결론 및 향후 연구에 대해 기술한다.

Ⅱ. 관련 연구

응용 프로그램 서비스 제공자는 방화벽을 우회하여 사용자에게 원활한 서비스를 제공하기 위해 복잡한 구조의 응용 레벨 프로토콜 구성하기 때문에 시그니쳐 또한 복잡하고 다양한 형태로 나타난다. 또한 인터넷에 기반한 응용의 증가로 인해 시그니쳐의 개수가 증가하고 그 가치 또한 높아지고 있다. 시그니쳐의 복잡도가 커지고, 개수가 증가하면서 페이로드 시그니쳐 기반 분류 시스템의 처리 속도는 트래픽 분류 시스템의 성능을 결정하는 중요한 요소로 작용하게 되었다.

분류 시스템의 처리 속도 향상을 위해 패턴 매칭 알고리즘의 성능 향상을 위한 다양한 방법을 제안하지만 매칭 알고리즘의 성능은 입력 데이터의 구성에 의존적이며, 제한적인 성능 향상을 나타낸다[4]. 패턴 매칭 알고리즘으로 오토마타에 기반 한 NFA와 DFA 알고리즘의 성능 향상을 위한 방법론들이 제시되고 있지만 오토마타를 이용한 방법은 '.*'와 같은 와일드 카드의 사용 빈도에 따라 시간 및 공간 복잡도 급격하게 증가하여 성능이 저하되는 문제점이 있다[5,6].

분석 시스템의 처리속도에 영향을 미치는 요소를 정의하고, 처리속도를 향상 시키는 최적의 분류 구조를 제시하는 연구에서는 입력 데이터의 탐색 공간을 최소화하기 위해 시그니쳐의 중복 제거와 , 시그니쳐를 계층적 구조화 하는 방법, 탐색 공간을 최적화를 위해 검사하는 패킷의 수를 제한하는 방법 등이 제시되었다[7,8].

기존의 연구는 페이로드 시그니쳐 기반 트래픽 분류 시스템의 처리 속도 향상을 위해서 패턴 매칭 기법을 소프트웨어 또는 하드웨어적으로 개선하려는 노력과 트래픽의 특징을 정의하고 그룹화 하는 방법이 주를 이루었다. 하지만 이러한 방법은 네트워크 대역폭 증가에 비해 상대적으로 제한적인 성능 향상을 보이거나 현재의 네트워크 환경에 적용하기에는 난해한 점들이 존재한다.

본 논문에서는 정확한 Offset을 추출하기 위해 Offset 추출 과정을 자동화한다. 또한, 시그니쳐 매칭 속도를 향상시키기 위하여 검사하는 패킷의 수, 전송 방향, 탐색 범위 등 자동화를 통해 추출된 다양한 Offset value를 활용하여 시그니쳐를 매칭 유형 별로

분류하고, 탐색 범위 최적화를 통해 페이로드 시그니쳐 기반 분석 방법의 성능을 향상 시키는 방법을 제안한다.

Ⅲ. 기존 시그니쳐 매칭 방법의 문제점

분석 시스템에서 단위 플로우에 모든 시그니쳐를 매칭시키는 풀 매칭 방식은 플로우에 다수의 시그니쳐가 매칭되는 중복과 충돌 매칭 현상이 불가피하게 발생한다는 단점과 매칭이 되어도 플로우의 끝까지 매칭을 수행한다는 단점이 있다[9]. 이러한 문제 때문에 일반적인 응용 분석 시스템에서는 플로우에 대해 응용 시그니쳐가 매칭 되면 매칭을 종료하는 부분 매칭(Partial-matching)방식을 사용한다. 그림 1은 일반적인 트래픽 분석 방법의 탐색범위를 나타낸다. 시그니쳐가 매칭 될 경우 페이로드의 처음부터 매칭 지점까지 탐색을 한다. 만약 시그니쳐가 매칭 되지 않는 경우 페이로드의 전체를 탐색하여 불필요한 탐색을 수행하고, 분석시간에 영향을 미친다.

기존 알고리즘의 불필요한 탐색에 의한 분석 성능 저하 문제를 해결하기 위해 snort와 같은 침입 탐지 시스템들은 Offset rule을 통해 전송 방향, Offset, Depth 등을 제한할 수 있는 방법을 제공한다. 하지만 snort에서 Offset rule을 사용하는 것은 필수가 아닌 선택이므로 사용자에 따라 시그니쳐의 탐색 범위가 각기 다를 수 있다. Offset을 추출하는 과정 역시 수동으로 작업하기 때문에 추출되는 Offset의 정확도나 작업시간 문제가 발생할 수 있다. 또한, 탐색하는 패킷의 수를 제한하거나, 전송 방향을 제한하는 등의 성능 향상을 이끌어 낼 수 있는 다양한 방법들을 잘 활용하지 못하고 있다.

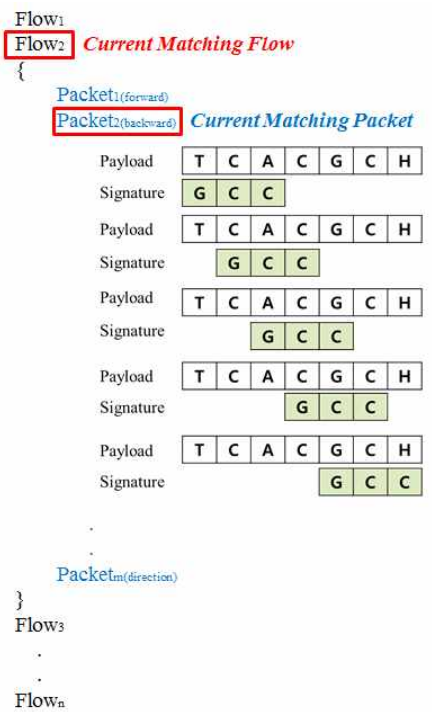

그림 1. 일반적인 트래픽 분석에서의 탐색범위
Fig. 1. Searching range of established traffic analysis

본 논문에서 제안하는 방법은 매칭 된 패킷이 플로우의 몇 번째 패킷인지를 알 수 있는 패킷의 순서 정보, 매칭 된 패킷이 서버로 향하는지 클라이언트로 향하는지에 대한 전송 방향 정보 그리고 시그니쳐가 페이로드 스트링의 어느 위치에 매칭 되었는지 알 수 있는 위치 정보 등 시그니쳐가 페이로드에 매칭 될 때 얻을 수 있는 다양한 Offset value를 추출하는 과정을 자동화하여 빠르고 정확히 추출한다. 또한, 추출한 Offset value를 활용하여 시그니쳐를 매칭 유형 별로 분류하고 시그니쳐의 탐색 범위를 최적화 한다. 탐색 범위의 최적화를 통해 불필요한 탐색 수행을 줄이고, 이를 통해 트래픽 분석 시스템의 속도 향상을 얻을 수 있다.

Ⅳ. 시그니쳐 매칭 유형 분류 방법

본 장에서는 시그니쳐의 매칭 되는 범위와 패킷의 순서, 방향성 등 다양한 Offset value를 자동 추출하는 방법과 추출된 Offset value를 활용하여 시그니쳐를 매칭 유형별로 분류하는 방법에 대해 기술한다.

그림 2는 전체 시스템의 구성도이다. 첫 단계에서는 트래픽 트레이스와 시그니쳐 리스트를 입력으로 받아 Offset 자동 생성기에서 시그니쳐 매칭 되는 패킷의 순서와 방향성, 그리고 범위 등 다양한 Offset value를 자동으로 추출하고, 두 번째 단계에서 추출한 Offset를 활용하여 시그니쳐를 매칭 유형 별로 분류한다. 마지막 세 번째 단계에서는 분류된 시그니쳐의 매칭 유형을 고려하여 트래픽을 분석한다.

4.1 Offset value

본 절에서는 시그니쳐를 매칭 유형 별로 6가지 유형으로 나누기 위한 알고리즘의 파라미터 값으로 사용되는 시그니쳐의 매칭 위치 정보(Matching Offset value)와 이를 자동으로 추출하는 알고리즘을 설명한다.

표 1은 시그니쳐를 매칭 유형 별로 분류하기 위해 사용되는 Offset value를 표로 정리한 것이다. 매칭 되

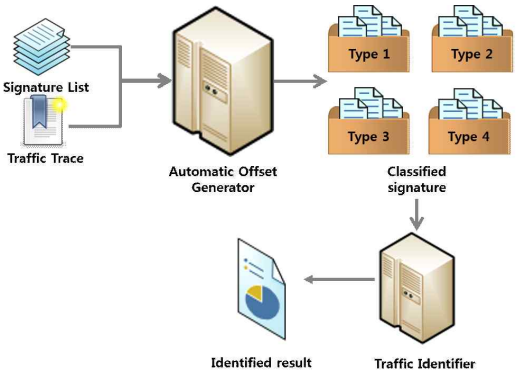

그림 2. 시그니쳐 매칭 유형 분류 시스템의 구성도
Fig. 2. Diagram of signature categorizing system

표 1. 유형 분류에 사용되는 Offset value의 정의
Table 1. Definition of Offset value used for signature categorizing

Offset value	explanation
Packet Offset	sequence value of packet when signature matching sequence of packet is fixed
Direction	value that packet's transmission direction is to-server or to-client when signature matched
First Offset	position value when starting position of signature matching is fixed
First Range	minimum position value when starting position of signature matching is not-fixed
Last Offset	last position value when signature matched
Depth	range information which signature is matched. (Last Offset minus First Offset or Last Offset minus First Range)

는 패킷의 순서가 fixed일 경우 사용하는 Packet Offset, 패킷의 전송 방향이 fixed일 경우 사용하는 Direction, 매칭 되는 Offset이 fixed일 경우 매칭의 시작 위치 값인 First Offset는 not fixed 경우는 사용하지 않는다. First Range는 매칭 되는 Offset이 not fixed일 경우의 매칭이 시작되는 값의 최솟값으로 First Offset이 not fixed인 경우에만 사용한다. Last Offset은 매칭이 끝난 지점들의 최댓값이고, Depth는 매칭이 끝난 지점에서 시작된 지점을 뺀 길이 값이다.

그림 3은 시그니쳐 매칭 위치 정보를 입력으로 받아 Offset value의 값을 결정하고 탐색 범위를 결정하는 알고리즘으로 Offset value 자동 추출에 사용된다. 시그니쳐의 Offset value를 자동 추출하기 위해 시그

```
    mS: Flow matched signature
    PO(mS) : Packet offset set of  mS
    Dir(mS) : Direction set of mS
    FO(mS) : first offset set of mS
    LO(mS) : Last offset set of mS
    Input : Signature info container
    Output : Signature matching range
1:  for each mSi do
2:    //first offset decision
3:    if | FO(mSi) | == 1
4:      FO(mSi) is fixed offset
5:    else
6:      find FOmin(mSi) value among FO(mSi)
7:    //Last offset decision
8:      find LOmax(mSi) value among LO(mSi)
9:    //Packet offset decision
10:   if | PO(mSi) | == 1
11:     mSi is matched only in PO(mSi) then,
12:     //range decision
13:     FOmin(mSi) ≤ mSi.Range ≤ LOmax(mSi)
14:   //Packet Direction decision
15:   else if | Dir(mSi) | == 1
16:     mSi is matched only in Dir(mSi) then,
17:     //range decision
18:     FOmin(mSi) ≤ mSi.Range ≤ LOmax(mSi)
19:   else
20:     //range decision
21:     FOmin(mSi) ≤ mSi.Range ≤ LOmax(mSi)
22: done
```

그림 3. Offset 자동 추출을 위한 탐색 범위 결정 알고리즘
Fig. 3. Searching range decision algorithm for automatic Offset extraction

니쳐가 각 테스트 셋 트레이스에 매칭되는 Offset 정보들을 활용한다. 우선 First Offset 값이 고정인지 확인하기 위해 시그니쳐가 매칭 된 시작 Offset 정보를 이용하여 판단한다. 시작 Offset 값이 유일하면 그 값을 First Offset으로 사용하고 First Offset이 고정이라고 판단하며, 두 개 이상일 경우 최솟값을 First Range로 사용한다. 다음으로 시그니쳐가 매칭 된 패킷의 위치정보를 확인한다. 패킷의 위치정보가 고정일 경우 그 값을 Packet Offset으로 사용하여 탐색시에 해당 패킷만을 탐색한다. 마지막으로 시그니쳐가 매칭 된 패킷의 방향성을 확인한다. 패킷의 방향성이 동일한 경우, 해당하는 방향성을 가지는 패킷만을 탐색한다.

그림 4는 시그니쳐 매칭 유형 정보와 Offset value를 포함하여 작성된 시그니쳐 모델의 예이다.

".*BitTorrent protocol.*" 시그니쳐는 결정된 시그니쳐 매칭 유형이 1번 유형이고, 패킷의 전송방향이 서버를 향하며, 시그니쳐가 매칭 되는 범위가 Offset 값 2부터 19만큼의 길이를 가진다고 해석한다.

```
payload=".*BitTorrent protocol.*"
Offset_Type="1" Direction="forward" First_Offset="2" Depth="19"
```

그림 4. Offset value를 포함한 시그니쳐 모델
Fig. 4. Signature model which contains Offset value

4.2 Offset value에 따른 유형 분류

표 2는 4.1장에서 설명한 6가지의 Offset value를

표 2. 시그니쳐 유형 결정 테이블
Table 2. Signature type decision table

Pattern Type	Packet Offset	Direction	First Offset	First Range	Depth
Type 1	O	X	O	X	O
Type 2	O	X	X	O	O
Type 3	X	O	O	X	O
Type 4	X	O	X	O	O
Type 5	X	X	O	X	O
Type 6	X	X	X	O	O

활용하여 시그니쳐의 매칭 유형을 결정하는 유형 결정 테이블이다. 타입은 총 6가지이며 크게 Packet Offset이 fixed인 유형1과 유형2, Direction이 fixed인 유형3과 유형4, 그리고 Packet Offset과 Direction이 모두 not fixed인 유형5와 유형6 3가지로 나뉜다.

그림 5는 Packet Offset이 fixed인 경우의 탐색범위이다. 매칭 되는 패킷의 순서 값이 fixed이기 때문에 각 플로우의 해당 순서 값 패킷만 탐색하면 된다.

Direction이 fixed인 유형 3,4번의 경우 탐색범위이다. 매칭 되는 패킷의 방향성이 fixed이기 때문에 각 플로우의 패킷 중 방향성이 Direction의 값과 일치하는 패킷만 탐색하면 된다.

유형 5, 6번은 Packet Offset과 Direction이 모두 not fixed이기 때문에 각 플로우의 모든 패킷을 탐색해야한다.

크게 나눈 3가지 유형은 다시 First Offset을 가지는 유형과 First Range를 가지는 유형으로 나눌 수 있다. First Offset을 가질 경우 매칭이 시작된 첫 바이트만 확인해도 매칭의 여부를 확인할 수 있다. First Range를 가지는 경우는 First Range부터 Depth의 길이만큼 탐색한다. 유형 번호가 빠를수록 탐색범위가 좁다.

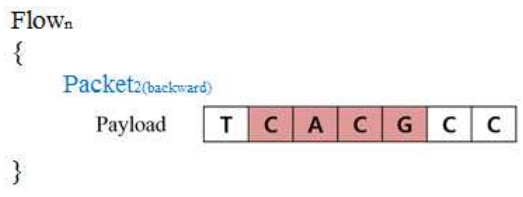

그림 5. 시그니쳐 매칭 유형 1, 2의 탐색범위
Fig 5. Searching range for signature matching type 1 and 2

V. 실 험

본 장에서는 4장에서 기술한 시그니쳐의 Offset value를 자동으로 추출하고, 추출한 Offset value를 활용하여 시그니쳐를 매칭 유형별로 분류하는 방법의 성능을 평가한다.

실험은 해당 트래픽 트레이스의 분석을 통해 자동으로 추출된 시그니쳐의 Offset value를 활용하여 시그니쳐를 매칭 유형별로 분류하고, 분류된 시그니쳐를 분석 시스템에 적용 시켜 기존의 트래픽 분석 시스템에 비해 얼마나 더 효율적인지 평가한다.

성능은 분석률과 분석시간 두 가지 척도를 기준으로 판단한다. 첫 번째 척도인 분석률은 트래픽 분석과정에서 시그니쳐를 매칭 유형별로 분류하고 탐색범위를 축소시킨 후에도 기존의 방법과 분석률이 동일한지를 판단하기 위해 사용한다. 두 번째 척도인 분석시간은 분석 시스템이 시그니쳐를 매칭 하는 과정에서 CPU가 실제로 작업을 수행하는데 걸린 시간인 User-time을 측정한다.

5.1 실험 환경

본 절에서는 시그니쳐의 매칭 유형 분류를 이용한 트래픽 분석의 성능 평가 실험을 위한 실험 환경에 대해 기술한다.

실험을 위해 수집한 트래픽은 모두 토렌트(Torrent) 트래픽으로 WireShark와 Net Monitor를 통해 트래픽

표 3. 실험에 사용된 Test set 정보
Table 3. Test set information which used in experiment

Traffic Trace ID	Content Size (MB)	Duration (min.)	Flow	Pkt	Byte
006_UT_FP02	119	24	10,874	2.1E+06	1.8E+09
006_UT_FP03	42.1	6	3,988	1.0E+06	8.4E+08
006_UT_FP04	355	7	2,996	1.5E+06	1.3E+09
006_UT_FP05	46.1	9	2,961	1.1E+06	9.7E+07
006_BT_FP02	119	18	4,946	2.0E+06	1.8E+09
006_BT_FP03	42.1	8	3,329	9.4E+05	8.4E+08
006_BT_FP04	355	10	3,603	1.5E+06	1.3E+09
006_BT_FP05	46.1	5	2,619	1.3E+05	1.2E+09
008_UT_FP06	20	1	278	4.8E+04	3.5E+07
008_UT_FP07	961	21	2,505	1.1E+06	1.2E+09
008_BT_FP08	899	5	1,760	5.3E+05	4.5E+08
008_BT_FP09	1,490	3	2,262	6.2E+05	6.8E+08
008_UT_FP10	1,520	1	1,363	1.0E+06	9.6E+08
008_UT_FP11	25	1	330	3.3E+04	2.8E+07
008_UT_FP13	954	5	3,204	1.0E+06	8.5E+08
Total	6693.4	124	47,018	1.46E+07	1.34E+10

표 4. 실험에 사용된 토렌트 페이로드 시그니쳐
Table 4. Torrent signatures used in the experiment

Sig ID	Payload signature	Protocol	Port
0	.*BitTorrent protocol.*	TCP	N/A
1	.*BitTorrent protocol.*	UDP	N/A
2	.*d1:ad2:id20.*	TCP	N/A
3	^d1:ad2:id20.*	UDP	N/A
4	.*d1:rd2:id20.*	TCP	N/A
5	^d1:rd2:id20.*	UDP	N/A
6	.*find_node.*UT.*	UDP	N/A
7	.*info_hash.*get_peer.*	UDP	N/A
8	.*5:peers.*	TCP	N/A
9	.*User-Agent:.*uTorrent.*	TCP	N/A
10	.*Host: com-utorrent.*	TCP	N/A
11	.*User-Agent: BTWebClient.*	TCP	N/A
12	.* //announce.info_hash=.*peer_id=.*	TCP	N/A
13	.*//scrape.info_hash=.*	TCP	N/A
14	.*d5:added.*	TCP	N/A
15	.*d5:added.*	UDP	N/A
16	.*Host:.*utorrent\.com*."	TCP	N/A
17	.*Host:.*bittorrent\.com*."	TCP	N/A
18	.*bittorrent.com.	UDP	N/A
19	.*tracker.*	UDP	N/A
20	*BT.*announce	UDP	N/A
21	.*UT.*announce.*	UDP	N/A
22	.*d2:ip6:.*1:rd2:id20.*	UDP	N/A
23	^d2:ip6:.*1:rd2:id20.*	UDP	N/A
24	^.....BT.....*	UDP	N/A
25	^.....UT.....*	UDP	N/A
26	.*\x00\x00\x04\x17\x27\x10\x19\x80\x00\x00\x00.*	UDP	N/A
27	^BT-SEARCH.*	UDP	N/A
28	.*BitTorrent.*	UDP	1900
29	.*uTorrent.*	UDP	1900
30	.*Host:.*deluge-torrent\.org.*	TCP	N/A
31	.*User-Agent: Deluge.*	TCP	N/A

을 모은 뒤 다른 응용의 트래픽은 모두 제외시키고, 토렌트 응용의 트래픽만을 수집했다. 수집된 트래픽은 총 15개의 트레이스이며, 성능 평가를 위한 Test set으로 사용 되었다. Testing Set의 정보는 표 3과 같다.

실험에 사용된 페이로드 시그니쳐의 개수는 총 32개로 표 4와 같다. 32개의 시그니쳐 모두 토렌트 응용 트래픽을 분석하기 위한 시그니쳐이다.

5.2 실험 결과

006_UT_FP02 트레이스를 표 3의 시그니쳐로 분석한 결과, 추출된 시그니쳐의 Offset value는 그림 6과 같다. 총 32개의 시그니쳐 중 14개의 시그니쳐가 해당 트레이스에서 매칭 되었고, 탐색 범위를 가장 축소 할 수 있는 유형 1이 5개, 유형2가 5 그리고 유형5가 4개로 분류 되었다.

시그니쳐의 매칭 유형별 분류를 적용시켜 테스트 셋 트래픽을 분석하여 기존의 트래픽 분석 방법과 비교한 결과를 표 5에 나타내었고, 그림 7에는 각각의 분석 방법으로 테스트 셋을 분석한 시간의 총합을 나

```
Sig ID[00] [Offset Type:  1] [PO:  1] [FO:    2] [DEPTH:   19]
Sig ID[03] [Offset Type:  5] [PO:  0] [FO:    1] [DEPTH:   11]
Sig ID[05] [Offset Type:  5] [PO:  0] [FO:    1] [DEPTH:   11]
Sig ID[07] [Offset Type:  5] [PO:  0] [FO:   35] [DEPTH:   84]
Sig ID[08] [Offset Type:  2] [PO:  2] [FR:   92] [DEPTH:  125]
Sig ID[09] [Offset Type:  2] [PO:  1] [FR:   50] [DEPTH:  152]
Sig ID[11] [Offset Type:  2] [PO:  1] [FR:   50] [DEPTH:   41]
Sig ID[12] [Offset Type:  1] [PO:  1] [FO:    6] [DEPTH:   72]
Sig ID[13] [Offset Type:  1] [PO:  1] [FO:    6] [DEPTH:   17]
Sig ID[16] [Offset Type:  2] [PO:  1] [FR:   24] [DEPTH:  724]
Sig ID[17] [Offset Type:  2] [PO:  1] [FR:   24] [DEPTH:  512]
Sig ID[23] [Offset Type:  5] [PO:  0] [FO:    1] [DEPTH:   23]
Sig ID[25] [Offset Type:  1] [PO:  1] [FO:    1] [DEPTH:   41]
Sig ID[26] [Offset Type:  1] [PO:  1] [FO:    1] [DEPTH:   12]
```

그림 6. 006_UT_FP02 트레이스의 Offset value 추출 정보
Fig. 6. Offset value information extracted from 006_UT_FP02 trace

표 5. 분석결과 비교
Table 5. Comparing of experimental result

Traffic Trace ID	not apply signature matching type categorizing		apply signature matching type categorizing	
	analysis rate(%)	analysis time(sec)	analysis rate(%)	analysis time(sec)
006_UT_FP02	94.64	0.59	94.64	0.49
006_UT_FP03	98.57	0.23	98.57	0.28
006_UT_FP04	99.52	0.18	99.50	0.14
006_UT_FP05	99.32	0.17	99.32	0.12
006_BT_FP02	89.77	0.31	89.77	0.24
006_BT_FP03	96.67	0.19	96.67	0.15
006_BT_FP04	98.36	0.21	98.36	0.17
006_BT_FP05	99.01	0.14	99.01	0.11
008_UT_FP06	85.61	0.02	85.61	0.02
008_UT_FP07	99.28	0.25	99.28	0.19
008_BT_FP08	85.11	0.12	85.11	0.10
008_BT_FP09	96.91	0.14	96.91	0.10
008_UT_FP10	84.23	0.08	84.23	0.07
008_UT_FP11	94.24	0.03	94.24	0.02
008_UT_FP13	85.96	0.22	85.96	0.17
Total	94	2.88	94	2.27

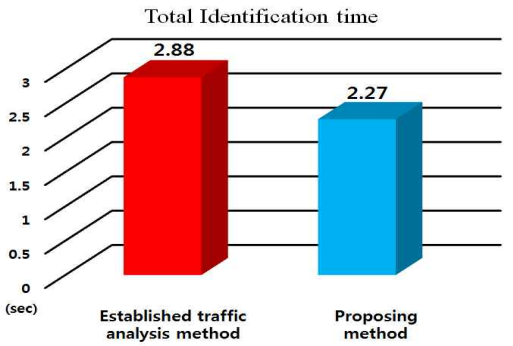

그림 7. 통합 분석시간 비교
Fig. 7. Total analysis time comparing

타내었다. 전체적으로 제안하는 방법은 기존의 트래픽 분석방법과 비교 했을 때, 분석률은 동일했으며, 분석

시간 측면에서는 평균적으로 약 25%의 분석 시간 단축을 보였고, 분석 시간 단축의 최댓값은 41%로 측정되었다.

결과적으로 시그니쳐의 매칭 유형을 분류하여 트래픽 분석에 적용한 것이 적용하지 않은 것과 비교 했을 때, 탐색 범위가 축소되었음에도 분석률이 동일한 것은 불필요한 탐색 범위만을 축소시켰다고 분석할 수 있다. 또한, 실험 트레이스마다 분석시간의 축소 폭이 차이 나는 부분은 해당 트래픽에 매칭 되는 시그니쳐와 시그니쳐의 매칭 유형이 각각 달라 축소된 탐색 범위가 달랐기 때문이다.

탐색 범위를 제한하는 Offset value 중 패킷의 순서를 제한하는 값인 Packet Order는 탐색 범위 축소에 매우 효과적이지만, 패킷 기반 분석을 하는 snort에서는 적용하기는 상대적으로 어려운 점이 있다. 페이로드 시그니쳐 기반 트래픽 분석의 속도 향상 측면에서 이런 다양한 Offset value를 효율적으로 사용할 수 있는 Flow 기반의 분석 방법이 요구 된다.

결과적으로 실험을 통해 본 논문에서 정의하고 제안하는 Offset 추출 과정의 자동화와 시그니쳐를 매칭 유형에 따라 분류하는 것이 효과적이라는 것을 증명하였다.

Ⅵ. 결론 및 향후 과제

본 논문에서는 Packet order, Direction 등 Offset rule보다 다양한 Offset value를 추출하는 과정을 자동화하고, 추출한 Offset value를 사용하여 시그니쳐를 매칭 유형별로 총 6 가지로 분류하여 분석하는 방법을 제안하였다.

제안하는 방법은 자동화를 통해 Offset value를 정확하고 빠르게 추출 하고, 탐색하는 패킷 순서, 패킷의 방향성 그리고 시그니쳐 Offset 제한과 매칭 유형별 분류 적용을 통해 분석률은 유지하면서 기존의 트래픽 분석 방법보다 최대 41%, 평균 25%의 분석 시간 단축을 보였다. 이를 통해 시그니쳐 매칭 유형 분류를 사용하지 않은 분석 방법에서 불필요한 탐색 범위만이 제거됨을 알 수 있었다. 또한, snort rule에서 제공하는 Offset rule의 사용은 페이로드 시그니쳐 기반 트래픽 분석에서 선택이 아닌 필수 사항이기 때문에 보완이 필요하고, 탐색 범위를 제한하기 위해 체계적인 시그니쳐 구조가 필요함을 알 수 있었다.

향후 연구로는 ".*"를 통해 여러 개의 싱글 시퀀스 시그니쳐들이 하나의 시그니쳐로 표현된 멀티 시퀀스 시그니쳐의 구조 정의에 대한 연구를 진행 할 계획이다.

References

[1] C.-S. Park, J.-S. Park, and M.-S. Kim, "Automatic payload signature generation system," *J. KICS*, vol. 38B, no. 08, pp. 615-622, Aug. 2013.

[2] J.-H. Choi, J.-S. Park, and M.-S. Kim, "Processing speed improvement of http traffic classification based on hierarchical structure of signature," *J. KICS*, vol. 39B, no. 04, pp. 191-199, Apr. 2014.

[3] F. Yu, Z. Chen, Y. Dino, T. V. Lakshman, and R. H. Katz, "Fast and memory efficient regular expression matching for deep packet inspection," in *Proc. ACM/IEEE Symp. Architecture Netw. Commun. Syst. (ANCS '06)*, pp. 93-102, San Jose, USA, Dec. 2006.

[4] C. L. Hayes and Y. Luo, "DPICO: A high speed deep packet inspection engine using compact finite automata," in *Proc. ACM/IEEE Symp. Architecture Netw. Commun. Syst. (ANCS '07)*, pp. 195-203, Orlando, USA, Dec. 2007.

[5] G. Vasiliadis, M. Polychronakis, S. Antonatos, E. P. Markatos, and S. Ioannidis, "Regular expression matching on graphics hardware for intrusion detection," in *Proc. 12th Int. Symp. Recent Advances Intrusion Detection (RAID '09)*, pp. 265-283, Saint-Malo, France, Sept. 2009.

[6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd Ed., MIT Press and McGraw-Hill, 2001.

[7] J.-S. Park, S.-H. Yoon, J.-W. Park, H.-S. Lee, S.-W. Lee, and M.-S. Kim, "Performance improvement of the payload signature based traffic classification system," *J. KICS*, vol. 35, no. 09, pp. 1287-1294, Sept. 2010.

[8] J.-S. Park, S.-H. Yoon, and M.-S. Kim, "Performance improvement of signature-based traffic classification system by optimizing the search space," *J. KSII*, vol. 12, no. 3, pp. 89-99, Jun. 2011.

[9] S.-H. Lee, J.-S. Park, M.-S. Kim, and W.-J. Seok, "Application traffic identification speed

improvement by optimizing payload signature matching sequence," *J. KICS*, vol. 40, no. 03, pp. 575-585, Mar. 2013.

정 우 석 (Woo-Suk Jung)

2015년 : 고려대학교 컴퓨터 정보학과 졸업
2015년~현재 : 고려대학교 컴퓨터정보학과 석사과정
<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석

박 준 상 (Jun-Sang Park)

2008년 : 고려대학교 컴퓨터 정보학과 졸업
2010년 : 고려대학교 컴퓨터 정보학과 석사
2010년~현재 : 고려대학교 컴퓨터 정보학과 박사과정
<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석, 트래픽 분류

김 명 섭 (Myung-Sup Kim)

1998년 : 포항공과대학교 전자계산학과 졸업
2000년 : 포항공과대학교 컴퓨터공학과 석사
2004년 : 포항공과대학교 컴퓨터공학과 박사
2006년 : Post-Doc. Dept. of ECE, Univ. of Toronto, Canada
2006년~현재 : 고려대학교 컴퓨터정보학과 부교수
<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석, 멀티미디어 네트워크