완전 자동화 페이로드 시그니쳐 업데이트 시스템

Fully Automatic Payload Signature Update System

심규석, 구영훈, 이성호, 김명섭

고려대학교 컴퓨터정보학과

{kusuk007, gyh0808, gaek5}@korea.ac.kr,

요 약

오늘날 네트워크 자원을 사용하는 응용이 증대되면서 네트워크 관리를 위한 트래픽 분석에서 현재 연구 단계의 한계가 드러나고 그런 한계를 해결하기 위한 다양한 연구가 진행되고 있다. 그 중 대표적인 연구인 시그니쳐 자동 생성 연구는 응용 트래픽을 입력으로 트래픽의 공통된 패턴을 찾아 출력하는 과정이 자동화된 연구이다. 그러나 시그니쳐 자동 생성 연구는 트래픽을 사용자가 수집해야하는 반자동 시스템이기 때문에 트래픽 수집 단계에서 문제가 발생하고, 생성된 시그니쳐의 정확도를 신뢰할 수 없는 한계가 있다. 본 논문에서는 시그니쳐 자동 생성 시스템의 한계를 극복하기 위해 트래픽 수집, 시그니쳐 생성/검증/관리까지 모든 과정이 자동으로 이루어지는 시스템을 제안한다. 제안하는 방법을 학내 망의 실제 트래픽에 적용하여 추출한 시그니쳐는 분석률은 유지하며, 오탐률을 0 로 만드는 효과를 보였다.

Keyword: Automatic, Signature, Update, Identifier, Verifier, Generation

1. 서론

네트워크 관리 분야에서 트래픽 분석은 점점 더 중요시 되어가고 있다. 네트워크 환경은 증대되고 있고, 그 속에서 통신하는 트래픽은 매우 다양한 형태를 가진다. 다양한 종류의 트래픽을 정리하고, 조절하여 네트워크 자원을 최대한 효율적으로 사용하고, 네트워크 사용자에게 서비스를 원활하게 제공하는 것이 네트워크 관리의 목적이다[6].

네트워크 관리에 있어 가장 중요한 분야는 네트워크 모니터링이다. 네트워크 모니터링은 네트워크에서 특정 응용 및 서비스가 발생량을 알아내어, 그에 맞는 관리 정책을 수립하는 것을 의미한다. 네트워크 모니터링을 위한 트래픽 분석은 사용자에게는 질 높은 네트워크 서비스를 제공받도록 하고, 제공자에게는 최소한의 네트워크 자원으로 최대한에 질 높은 서비스를 제공할 수 있는 기반이 된다.

트래픽 분석에서 필수적으로 사용되는 것은 각 응용 별로 트래픽을 분류할 수 있는 시그니쳐 이다. 시그니쳐는 트래픽의 특징 별로 다양한 종류가 존재한다. 시그니쳐를 빠르고 정확하게 생성하기 위해 시그니쳐 자동 생성 연구는 활발하게 진행되고 있다. 시그니쳐 자동 생성 연구는 패킷의 페이로드 내용을 기반으로 공통 문자열을 자동으로 추출하여 시그니쳐화 하는 방법이다. 그러나 현재 단계의 시그니쳐 자동 생성 방법은 한계가 존재한다. 먼저, 시그니쳐를 만들기 위한 최소 조건으로 추출하고자 하는 응용의 트래픽을 수집 해야하는데 이 단계는 사용자가 직접 트래픽 수집 도구[4,5]를 이용하여 수집해야한다. 본 단계에서 트래픽을 잘 못 수집하면 추출된 시그니쳐는 잘못된 시그니쳐이다. 두번째는 수집된 트래픽이 단기간의 트래픽이기 때문에 시그니쳐 또한 응용의 대표적인 시그니쳐가 아닐 확률이 높다. 세번째는 시그니쳐 검증단계를 포함하지 않기 때문에 추출된 시그니쳐가 다른 응용을 분석할 확률이 높다. 마지막으로 항상 최신 시그니쳐를 유지할 수 없는 단점이 있다. 트래픽 패턴이 변화되더라도 인지하는 것은 여전히 사람이기 때문에 즉각적인 대응이 불가하다.

따라서 본 논문에서는 이러한 한계를 극복하기 위해 완전 자동화 시그니쳐 업데이트 시스템을 제안한다. 제안하는 시스템은 트래픽 수집, 시그니쳐 생성, 생성된 시그니쳐 검증 그리고 시그니쳐 관리까지 모든 과정이 자동으로 이루어 지는 시스템이다. 또한 지속적으로 시스템을 수행하기 때문에 항상 최신 시그니쳐를 유지할 수 있고, 트래픽 패턴 변화에 대해 즉각적인 대응이 가능하다.

본 논문은 1 장 서론에 이어, 2 장에서 시그니쳐 자동 생성 시스템에 대한 관련 연구에 대해 언급하고, 3 장에서 제안하는 시스템을 제안한다. 4 장에서는 제안한 시스템의 성능을 평가하고 마지막 5 장에서 결론 및 향후 연구에 대해 서술하고 논문을 마친다.

이 논문은 2015 년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No.2015R1D1A3A01018057)

2. 관련 연구

트래픽 분석을 위한 시그니쳐는 서론에서 언급했듯이 트래픽 특성에 따라 다양한 형태로 존재한다. 트래픽의 포트번호를 이용하여 분석하는 포트 기반 시그니쳐와 트래픽의 크기, 위치, 시간 등 통계적인 정보를 이용한 통계 기반 시그니쳐, 패킷의 데이터 부분인 페이로드 정보를 이용한 페이로드 기반 시그니쳐가 대표적이다.

포트 기반 시그니쳐는 IANA 에서 지정한 포트 정보를 시그니쳐로 사용한다. 본 방법은 적은 메모리 사용으로 매우 빠르게 트래픽 분석이 가능하지만, 현재 많은 응용들은 방화벽 및 IPS 장비를 통과하기 위해 임의 포트 번호를 사용하기 때문에 더 이상 포트 기반 시그니쳐는 무의미하다.

통계 기반 시그니쳐는 플로우 내의 패킷의 크기, 위치, 방향, 지속시간등을 시그니쳐로 사용한다. 본 방법은 암호화된 트래픽을 분석할 수 있고, 분석 속도가 빠른 장점이 있지만, 시그니쳐를 생성하는 것이 어렵고 응용이 한정되어 있을 뿐만 아니라 정확성을 기대하기 힘들다는 단점이 존재한다.

따라서 본 연구에서는 페이로드 기반 시그니쳐를 다룬다. 페이로드 기반 시그니쳐는 패킷의 데이터 부분인 페이로드 내의 공통된 문자열을 의미한다. 가장 정확도가 높은 시그니쳐이지만 시그니쳐 추출이 어렵고, 추출 과정에서 인적, 시간적 소비가 크다는 단점이 있다.

이러한 단점을 해결하기 위해 시그니쳐 자동 생성 방법이 연구되고 있다. 시그니쳐 자동 생성을 위해 다양한 알고리즘이 사용되고 있는데, 대표적으로 LCS (Longest Common String)알고리즘, Smith-Waterman 알고리즘과 가장 최근 연구에서 순차 패턴 알고리즘의 한 종류인 AprioriAll 알고리즘을 이용한 시그니쳐 자동 생성 연구가 있다.

LCS 알고리즘을 응용 트래픽 시그니쳐 추출 목적에 맞게 변형한 대표적인 방법은 LASER(LCS-based Application Signature ExtRaction)이다[1]. 본 방법은 두 개의 스트링을 비교하는 Matrix 에서 Backtracking 을 이용하여 연속된 공통 문자열을 찾는 방법이다. 따라서 두 개의 스트링을 계속 비교하여야 하기 때문에 추출 과정의 시간이 오래 걸리는 단점이 있다.

Smith-Waterman 은 본래 DNA 의 유사도를 판단하는 목적에 발표된 알고리즘이다[2]. 본 알고리즘을 사용한 응용 시그니쳐 자동 생성 방법도 발표되었는데 본 방법은 LCS 와 매우 유사하다. 하지만 Backtracking 방법에서 LCS 알고리즘은 연속된 공통 문자열을 찾을 수 있지만, Smith-Waterman 알고리즘은 연속된 공통 문자열의 집합을 찾을 수 있는 차이가 있다. 그러나 본 방법 또한 두 개의 스트링을 비교하는 것에 차이가 없기 때문에 추출 과정에 많은 시간이 소비된다.

가장 최근 연구인 AprioriAll 알고리즘을 이용한 시그니쳐 자동 생성 방법은 위의 단점을 해결할 수 있는 방법이다[3]. 위의 방법들은 특정 두 문자열을 비교하여 실제 트래픽에 적용하기 위해 트래픽의 순서를 정하거나, 그룹화시키는 전처리 과정과 생성된 부분 문자열을 하나의 규칙으로 통합시키는 후처리 과정이 필요하다면, 본 방법은 모든 문자열을 후보로 길이 1 부터 증가시키며 시그니쳐가 될 수 있는 가능성이 높은 문자열만 취하기 때문에 추출 과정에 많은 시간이 소비되지 않고, 전처리 과정과 후처리 과정이 필요 없는 장점이 있다.

3. 문제 정의

트래픽 분석이 다양한 이유로 매우 어려워지고 있다. 그 중 두가지의 가장 큰 이유가 있는데, 첫 번째로 지속적으로 변하는 트래픽 패턴이다. 트래픽은 일정한 패턴을 가지고 통신을 하게 된다. 하지만 서비스를 제공하는 업체에서 보안 위험에 예방하여 트래픽 패턴을 변화시킬 수 있다. 트래픽 패턴이 변화되면, 네트워크 관리자는 다시 트래픽 패턴을 분석하고, 그렇지 않다면 해당 서비스를 관리하지 못하게 된다. 두 번째는 새로운 응용이 급격하게 생성되고 많이 사용되고 있다. 모바일 앱 다운로드는 전 세계 시장에서 5 년 사이에 약 10 배정도 증가한 것으로 나타나고, 지속적으로 증가할 것이 예상된다. 네트워크 관리자는 이러한 새로운 응용에 대해 모두 다 파악하기 힘들고 파악한다 하더라도 모든 응용들을 분석하기는 한계가 있다.

이러한 한계를 극복하기 위해 시그니쳐 자동 생성의 연구는 활발히 이루어 지고 있다. 그러나 현재 단계의 시그니쳐 자동 생성은 트래픽 수집과 추출된 시그니쳐에 대한 검증, 그리고 시그니쳐 관리 방법에 대한 한계가 존재한다. 최적의 방법으로 시그니쳐를 자동 생성할 수 있지만 입력된 트래픽에서 문제가 발생하면 잘못된 시그니쳐가 추출될 수 밖에 없다. 또한 추출된 시그니쳐는 해당 응용만을 분석할 수 있다는 신뢰가 없고, 계속해서 추출되는 시그니쳐와 사용되지 않는 시그니쳐를 관리해야하는 한계는 존재한다. 따라서 본 논문에서는 이러한 과정을 완전 자동화 할 수 있는 완전 자동화 페이로드 시그니쳐 업데이트 시스템을 제안한다.

4. 완전 자동화 시그니쳐 업데이트 시스템

본 장에서는 완전 자동화 페이로드 시그니쳐 업데이트 시스템을 제안한다. 제안하는 시스템은 트래픽 수집, 시그니쳐 생성, 시그니쳐 검증, 그리고 시그니쳐 관리의 모든 과정이 자동으로 수행된다. 따라서 본 장에서 각 과정의 수행 과정에 대한 방법론을 설명한다. 그림 1 은 완전 자동화 페이로드 시그니쳐 업데이트 시스템의 수행 과정이다.

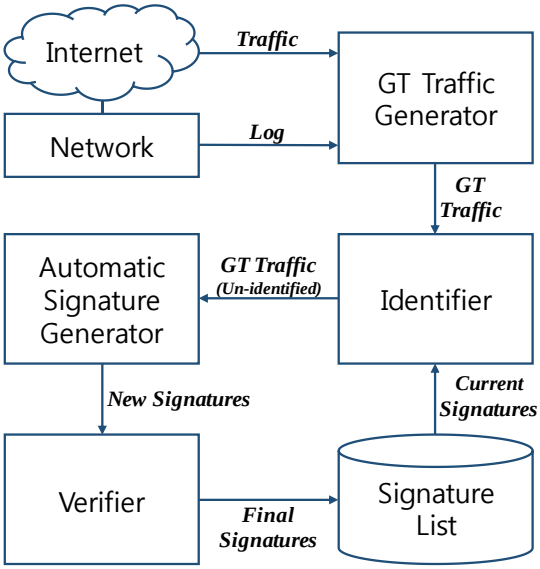

그림 1. 완전 자동화 시그니쳐 업데이트 시스템

그림 1 에서 GT Traffic Generator 에서는 트래픽 데이터와 호스트에서 발생하는 Log 데이터를 이용하여 응용 별 정답지 트래픽을 생성한다. Identifier 에서는 응용 별 정답지 트래픽과 기존 시그니쳐를 이용해 분석되지 않은 응용 별 정답지 트래픽을 출력한다. Automatic Signature Generator 에서는 분석되지 않은 응용 별 정답지 트래픽을 입력으로 새로운 시그니쳐를 출력한다. 새로운 시그니쳐는 Verifier 를 통해 검증된 시그니쳐를 분별하여 다시 Signature List 로 입력된다. 이러한 과정이 지속적으로 이어지면서 검증된 Final Signature 는 다음 수행과정 때 Current Signature 로 지속적인 검증 단계를 거쳐 항상 최신 시그니쳐를 유지할 수 있다.

4-1. GT Traffic Generator: 자동 트래픽 수집 및 정답지 트래픽 생성

기존 연구가 가진 가장 큰 한계는 트래픽을 사용자가 직접 수집해야하는 것이다. 이러한 과정에서 잘못된 트래픽이 수집될 수 있기 때문에 제안하는 시스템의 GT Traffic Generator 에서는 트래픽을 자동으로 수집하고, 각 응용 별로 정답지 트래픽을 생성한다. 정답지 트래픽을 생성하기 위해 본 시스템에서는 TMA(Traffic Measurement Agent)를 사용한다. TMA 는 각 호스트에서 실행되며 로그데이터를 남긴다. 표 1 은 TMA 에서 발생하는 로그데이터가 포함하는 정보이다.

표 1. TMA 정보

Process name
IP address (local, remote)
Port number (local, remote)
State (start, continue, end, server)
Protocol
Path

표 1 과 같은 정보들을 TMA 는 각 호스트에서 시간대별로 TMS(Traffic Measurement Server)로 전송한다. TMS 는 각 호스트에서 받아온 정보를 통합한다. 본 시스템에서는 TMS 로부터 통합된 정보와 트래픽의 정보를 매칭한다. 같은 시간, 같은 5-tuple(srcIP/Port, dstIP/Port, Protocol)를 가진 정보를 매칭하여 트래픽을 Process name 별로 저장하며 응용별 정답지 트래픽을 생성한다.

4-2. Identifier: 트래픽 분석 및 시그니쳐 관리

자동으로 수집된 정답지 트래픽을 이용하여 시그니쳐를 바로 생성할 수 있다. 하지만 같은 응용에 대해 지속적으로 동일한 시그니쳐가 추출될 수 있기 때문에 시스템에 불필요한 부하와 시간이 소비될 수 있다. 또한 기존 시그니쳐에서 트래픽 패턴 변화로 인해 사용되지 않는 시그니쳐에 대한 관리 방법이 필요하다. 본 시스템의 Identifier 에서는 기존 시그니쳐로 1 차 트래픽을 분석하여 분석되지 않는 트래픽을 분류하고, 사용되지 않는 시그니쳐를 삭제하며 시그니쳐를 관리한다.

최신 시그니쳐를 유지 하기 위해서는 새로운 시그니쳐를 생성할 뿐만 아니라 사용되지 않는 시그니쳐를 삭제해야한다. 이러한 과정이 생략된다면 시그니쳐는 지속적으로 축적되고, 축적된 시그니쳐는 트래픽 분석에 있어 시스템 부하 및 과도한 시간 소비의 원인이 된다. 따라서 본 시스템에서는 사용되지 않는 시그니쳐를 삭제한다. 수식(1)은 시그니쳐의 구성을 나타낸다.

$$\text{Signature} = \{\text{Header}, \text{Contents}, \text{Weight}, \text{Score}\} \quad (1)$$

다음과 같이 시그니쳐는 응용 서버의 정보인 IP address, Portnumber, 그리고 Protocol 로 이루어진 Header, 트래픽의 고유한 패턴인 Contents, 시그니쳐 삭제를 위한 가중치 값인 Weight, Weight 를 계산하기 위한 Score 값으로 구성된다. Score 는 알고리즘 1 과 같이 계산된다.

Procedure: *Calculation of Score*
Input: *Current Signatures, total GT traffic*
Output: *Cumulative Score by Signatures*

```
1: foreach signature S in OldSignatureSet do
2:   foreach flow F in GTtrafficSet do
3:     if ( identified(S,F) == 1 ) then
4:       S.CumulativeScore = 0; break;
5:     end
6:   end
7:   S.CumulativeScore ++;
8: end
```

알고리즘 1. Score 계산 알고리즘

다음과 같이 Score 는 분석에 사용되지 않은 횟수를 나타낸다. 그러나 Score 가 누적되더라도 분석에 사용된 시그니쳐는 다시 Score 값이 0 으로 초기화된다. 수식(2)는 Score 를 이용한 Weight 계산방법이다.

$$\mathrm{Weight_t}(S) = Weight_{t-p}(S) + C(S) - U_t(S) \quad (2)$$
$$(\mathrm{t = current\ time, p = period, C = completeness})$$

$$\mathrm{U_t}(S) = \left\lceil \frac{Weight_{(t-p)}(S)}{10} \right\rceil \times Score \quad (3)$$
$$\left(\mathrm{Weight_0} = 0, Weight \le 0, \left\lceil \frac{Weight(S)}{10} \right\rceil \ge 1\right)$$

시그니쳐에 Weight 를 구성한 것은 과거 분석률에 상당한 영향을 미친 시그니쳐이거나, 오랜 기간 분석에 사용되었던 시그니쳐가 삭제되기까지 시간을 주기 위함이다. 반면 기존 사용자가 잘못된 시그니쳐를 가지고 있었다면 Weight 값이 적어서 신속하게 삭제된다.

4-3. Automatic Signature Generator: 시그니쳐 자동생성

시그니쳐 자동 생성은 시그니쳐를 추출하기 위해 순차 패턴 알고리즘 중 Aprioriall 알고리즘을 수정하여 사용한다. 시그니쳐를 자동으로 생성하는 과정은 먼저 트래픽의 페이로드를 추출하여 시퀀스를 생성한다. 생성된 시퀀스들의 집합에서 길이 1 의 콘텐츠를 생성한다. 길이 1 콘텐츠는 최소 지지도 검사를 통해, 길이 2 로 생성된 후보자 콘텐츠와 삭제될 콘텐츠로 구분된다. 더 이상 길이가 증가되지 않을 때까지 이러한 과정을 반복하여 공통 문자열을 추출한다.

생성되는 시그니쳐는 그림 2 와 같이 총 3 가지 타입이 존재한다. 첫 번째 타입은 공통적으로 발생하는 연속된 문자열을 의미하는 콘텐츠 시그니쳐이다. 두 번째 타입은 동일한 패킷에서 발생하는 콘텐츠 시그니쳐의 조합을 의미하는 패킷 시그니쳐이다. 세 번째 타입은 동일한 플로우에서 발생하는 패킷 시그니쳐의 조합을 의미하는 플로우 시그니쳐이다.

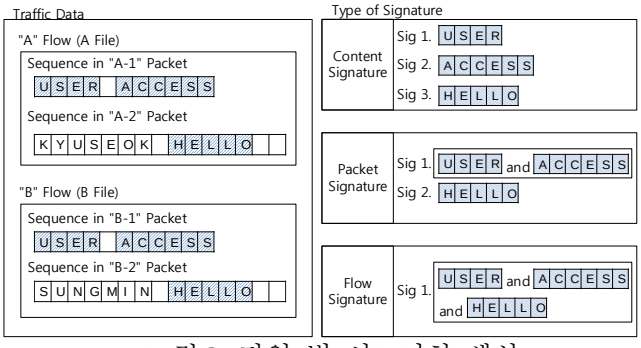

그림 2. 타입 별 시그니쳐 생성

4-4: Verifier: 시그니쳐 검증

제안하는 시스템은 새롭게 추출된 시그니쳐에 대해 검증 단계를 포함하고 있다. 검증단계에서는 "A" 응용 트래픽에 의해 추출된 시그니쳐가 "A" 응용이 아닌 "B","C"등 다른 응용 트래픽을 분석하는 것을 방지한다. 다른 응용 트래픽을 분석하는 시그니쳐는 시그니쳐로써의 의미를 잃어버리기 때문에 삭제해 주지만 아주 소량의 다른 응용 트래픽을 분석하고, 해당 응용 트래픽 분석에 큰 비중을 차지하고 있는 시그니쳐를 남기는 방법론을 제안한다.

시그니쳐를 검증은 다른 응용을 분석하는 시그니쳐 즉, False-Positive 가 있는 시그니쳐를 대상으로 한다. False-Positive 가 없는 시그니쳐는 최종 시그니쳐에 포함된다. 다음과 같이 검증과정에서 False-Positive 수치를 사용하게 된다. 표 2 는 데이터 판별 용어에 대해 설명한 것이다.

TP(True-Positive)는 A 응용 시그니쳐가 A 응용 트래픽을 분석한 수치이다. FN(False-Negative)는 A 응용 시그니쳐가 A 응용 트래픽을 분석하지 못한 수치이다. FP(False-Positive)는 A 응용 시그니쳐가 A 응용 트래픽을 제외한 나머지 트래픽을 분석한 수치이다. TN(True-Negative)는 A 응용 시그니쳐가 A 응용 트래픽을 제외한 나머지 트래픽을 분석하지 않은 수치이다.

표 2. 데이터 판별

	Relevant Traffic	Others Traffic
Analyze	TP(True-Positive)	FP(False-Positive)
Not Analyze	FN(False-Negative)	TN(True-Negative)

본 시스템에서 가장 중점적으로 FP(False-Positive) 수치를 다룬다. 특정 응용에 대한 시그니쳐 전체를 분석할 경우 TP(True-Positive)도 중요하지만, 현재 단계에서 각 시그니쳐 하나의 수치를 계산하기 때문에 FP 를 중점으로 다룬다. 각 시그니쳐 별 FP 를 계산하여 FP 가 0 인 시그니쳐는 최종 시그니쳐로 저장되고, FP 가 0 을 초과하는 시그니쳐에 한해 검증 단계에 입력된다. 시그니쳐의 검증을 위해 본 논문은 Precision, Recall, 그리고 F-measure 값을 사용한다. Precision 은 시그니쳐로 분석된 트래픽 중에 정확히 분석한 비율을 나타내고, Recall 은 특정 응용 트래픽 중 시그니쳐로 정확히 분석한 비율을 나타낸다. 다음과 같이 수식(4), 수식(5)이 표현한다.

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \quad (4)$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \quad (5)$$

F-measure 는 Precision 과 Recall 을 이용하여 가중치를 주어 값을 측정하는 공식이다. 본 논문에서는 F-measure 를 사용하여 Precision 에 가중치를 두기 위해 β값을 사용하는데 1 을 기준으로 1 보다 작으면 Precision 에 민감한 수식이 되고, 1 보다 크면 Recall 값에 민감한 수식이 된다. 만약 동일한 가중치를 주어야한다면 β를 1 로 고정하면 된다. 본 논문에서는 F-measure 를 사용하여 Precision 에 가중치를 두기 위해 β값을 0.1 로 고정하여 사용한다. 수식(6)은 F-

measure 표현식이다.

$$\mathrm{F-measure} = \frac{(\beta^2+1)\times Precision\times Recall}{\beta^2\times Precision\times Recall} \quad (6)$$

F-measure 의 최대값은 1 이고 최소값은 0 이다. 1 이 나오는 경우는 해당 시그니쳐가 해당 정답지 트래픽에 있는 모든 트래픽을 분석하고, 그 외 트래픽을 하나도 분석하지 못했을 때이고, 0 이 나오는 경우는 TP=0 가 되면 된다. 따라서 본 논문에서는 F-measure 가 최소 0.95 이상의 정확도를 가진 시그니쳐만을 최종 시그니쳐로 저장한다.

제안하는 방법을 통해 관리되는 네트워크에서 호스트가 사용하는 응용에 대한 시그니쳐는 지속적으로 업데이트가 되며, 사용되지 않은 시그니쳐는 삭제되고, 새로운 시그니쳐는 추출된다. 새로운 시그니쳐는 검증과정을 거치면서 정확도 높은 시그니쳐를 유지한다.

5. 실험

본 장에서는 완전 자동화 페이로드 시그니쳐 업데이트 시스템에 의해 생성되는 시그니쳐와 기존 시그니쳐 자동 생성 시스템에 의해 생성되는 시그니쳐의 분석률과 오탐률을 비교실험한다. 분석률은 TP(True-Positive)로써 해당 응용 트래픽 중 분석한 트래픽의 양으로 표현하고, 오탐률은 FP(False-Positive)로써 해당 응용을 제외한 트래픽 중 분석한 트래픽의 양으로 표현한다. 표 3 은 호스트에서 자주 사용하는 5 가지 응용을 선정하여 비교 실험한 결과이다.

본 실험결과에서 모든 응용에 대해 FP 수치를 0 으로 감소시켰음에도 불구하고, TP 수치는 많이 감소하지 않았다. 특히, Dropbox 의 경우 FP 수치가 0 으로 감소하였지만 TP 수치는 유지하는 효과를 나타냈다. 따라서 본 시스템을 통해 시그니쳐의 정확도를 향상시키고 최신 시그니쳐로 유지할 수 있었다.

표 3. 시그니쳐 정확도 비교 실험 결과

	미 검증 시그니쳐			검증 시그니쳐		
	TP	FP	개수	TP	FP	개수
Naver	2,068 /2,128	904 /3,582	542	1,929 /2,128	0 /3,582	508
Youtube	642 /645	2,458 /5,065	112	417 /645	0 /5,065	100
uTorrent	2,138 /2,613	2,591 /3,097	631	2,108 /2,613	0 /3,097	608
Dropbox	31 /51	5 /5,659	5	31 /51	0 /5,659	4
Facebook	273 /273	1,344 /5,437	37	212 /273	0 /5,437	22

다음 실험결과는 각 응용 시그니쳐들의 분석률(TP) 및 오탐률(FP)가 지속적으로 업데이트 됨에 따라 변화량을 표현한다. 기존 연구와 비교하기 위해 각 응용의 초기 시그니쳐는 시그니쳐 자동 생성 시스템에 의해 추출된 시그니쳐를 사용한다. 실험결과에서와 같이 초기 시그니쳐로 각 응용을 분석했을 때, FP 의 수치가 높은 것을 확인할 수 있지만 본 시스템인 시그니쳐 업데이트 시스템에 의해 추출된 시그니쳐로 분석한 결과인 두번째부터 FP 의 수치가 급격히 줄어든 것을 확인할 수 있었다. 그림 3 은 Naver 응용의 시도 횟수에 따른 분석률 및 오탐률이다.

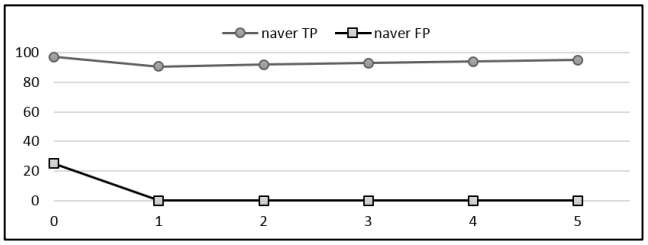

그림 3. 시그니쳐 업데이트 시도 횟수에 따른 naver 응용의 분석률 및 오탐률

6. 결론 및 향후 연구

본 논문에서 제안한 완전 자동화 페이로드 시그니쳐 업데이트 시스템은 기존 시그니쳐 생성 방법의 한계를 극복하기 위해 시그니쳐를 추출하기 위한 모든 과정인 트래픽 수집, 시그니쳐 생성, 시그니쳐 검증, 시그니쳐 관리까지 일련의 과정을 자동화한다. 제안된 시스템은 실험결과로 성능 향상이 증명되었다.

향후 본 시스템을 실시간에 실행하기 위한 성능 향상에 대한 연구가 필요하다. 또한 시그니쳐 자동 생성 과정에서 최적화된 순차 패턴 알고리즘을 선정해야한다.

참고문헌

[1] B.-C. Park, Y. J. Won, M.-S. Kim, and J. W. Hong, "Towards automated application signature generation for traffic identification," in *Network Operations and Management Symposium, 2008. NOMS 2008. IEEE*, 2008, pp. 160-167.

[2] X. Feng, X. Huang, X. Tian, and Y. Ma, "Automatic traffic signature extraction based on Smith-waterman algorithm for traffic classification," *in Broadband Network and Multimedia Technology (IC-BNMT), 2010 3rd IEEE International Conference on*, 2010, pp. 154-158..

[3]심규석, 윤성호, 이수강, 김성민, 정우석, 김명섭, "네트워크 트래픽 분석을 위한 snort content 규칙 자동 생성", 통신학회 논문지 Vol.40 No.04, pp.666-677, April. 2014.

[4]https://www.wireshark.org

[5]https://www.tcpdump.org

[6]Y. Wang, Y. Xiang, W. L. Zhou, and S. Z. Yu, “Generating regular expression signatures for network traffic classification in trusted network management,” *J. Netw. Comput. Appl.*, vol. 35, pp. 992-1000, May 2012.