http://dx.doi.org/10.7840/kics.2015.40.11.2160

SSL/TLS 기반 암호화 트래픽의 서비스 식별 방법

김 성 민◆, 박 준 상*, 윤 성 호**, 김 종 현***, 최 선 오****, 김 명 섭°

Service Identification Method for Encrypted Traffic Based on SSL/TLS

Sung-Min Kim◆, Jun-Sang Park*, Sung-Ho Yoon**, Jong-Hyun Kim***, Sun-Oh Choi****, Myung-Sup Kim°

요 약

네트워크 트래픽이 복잡, 다양해짐에 따라 발생하는 네트워크 보안문제 해결을 위해 다양한 암호화 프로토콜 중 하나인 SSL/TLS가 널리 사용되고 있다. 하지만 현재의 트래픽 분석 시스템은 암호화 트래픽을 프로토콜 레벨에 한정적으로 분석하고 있는 실정이다. 효과적인 네트워크 자원 관리를 위해서는 암호화 트래픽에 대한 서비스 단위 분석이 요구된다. 본 논문에서는 SSL/TLS 암호화 응용 트래픽의 페이로드 시그니쳐를 자동으로 추출하고, 이를 바탕으로 네트워크 트래픽 상에서 SSL/TLS 응용 서비스를 식별하는 방법을 제안한다. 이는 암호화 세션이 맺어질 때 초기에 발생하는 SSL/TLS Handshake 중 인증서 교환 레코드의 인증서 발행 대상정보를 시그니쳐로 이용하여 서비스를 식별을 하는 것이다. 본 논문에서 제안하는 방법은 95%에 가까운 SSL/TLS 트래픽을 분석 하였으며, 이 때 추출한 시그니쳐를 별도의 트래픽 트레이스에 적용시켜 각 서비스 별로 최대 95%의 정확도를 내어 그 성능과 가능성을 증명하였다.

Key Words : SSL/TLS, Payload Signature, Handshake, Certificate, Traffic Classification

ABSTRACT

The SSL/TLS, one of the most popular encryption protocol, was developed as a solution of various network security problem while the network traffic has become complex and diverse. But the SSL/TLS traffic has been identified as its protocol name, not its used services, which is required for the effective network traffic management. This paper proposes a new method to generate service signatures automatically from SSL/TLS payload data and to classify network traffic in accordance with their application services. We utilize the certificate publication information field in the certificate exchanging record of SSL/TLS traffic for the service signatures, which occurs when SSL/TLS performs Handshaking before encrypt transmission. We proved the performance and feasibility of the proposed method by experimental result that classify about 95% SSL/TLS traffic with 95% accuracy for every SSL/TLS services.

※ 이 논문은 2015년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원(No.B0101-15-0300, 사이버 공격의 사전 사후 대응을 위한 사이버 블랙박스 및 통합 사이버보안 상황분석 기술 개발) 및 2013년 BK21 플러스 사업(No. T1300572) 및 2015년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2015R1D1A3A01018057).
◆ First Author : Dept. of Computer and Information Science, Korea University. gogumiking@korea.ac.kr, 학생회원
° Corresponding Author : Dept. of Computer and Information Science, Korea University. tmskim@korea.ac.kr, 종신회원
* Dept. of Computer and Information Science, Korea University. junsang_park@korea.ac.kr, 학생회원
** Dept. of Computer and Information Science, Korea University. sungho_yoon@korea.ac.kr, 학생회원
*** Network Security Research Section, Cyber Security Research Laboratory, ETRI. suno@etri.re.kr
**** Network Security Research Section, Cyber Security Research Laboratory, ETRI. jhk@etri.re.kr
논문번호 : KICS2015-03-045, Received March 5, 2015; Revised June 15, 2015; Accepted November 4, 2015

Ⅰ. 서 론

오늘날 초고속 인터넷 보급과 네트워크 장비의 발달로 인해 발생하는 트래픽이 복잡, 다양해지고 있기 때문에, 네트워크의 효과적인 운용과 관리를 위해서는 네트워크 트래픽 분석은 필수적인 요소이다. 또한 개인정보 유출, 사생활 침해, 계정 도용 등 네트워크 보안 문제가 심각해지고 있다. 이러한 보안문제 해결을 위해 다양한 암호화 프로토콜이 개발되었는데, 대표적으로 SSL/TLS[1], SSH[2] 등이 있다. 하지만 보안 이슈가 대두됨에 따라 Facebook, Google 등 암호화 트래픽이 증가되고 있으나 응용 서비스 단위의 분류가 아닌, SSL/TLS와 같은 암호화 프로토콜 단위 또는 암호화 여부에 대한 분류가 주를 이루고 있다. 네트워크 트래픽의 서비스 분류를 통한 효과적인 네트워크 자원 관리를 위해서는 암호화 트래픽에 대한 서비스 단위 분석이 요구된다.

따라서 본 논문에서는 SSL/TLS 암호화 프로토콜을 사용하는 네트워크 응용을 다양한 트래픽 분류방법 중 페이로드 시그니쳐에 기반 한 분류 방법[3]을 이용하여 암호화 트래픽을 서비스 단위로 식별하는 방법을 제안한다.

본 논문은 다음과 같은 순서로 기술한다. 2장에서는 기존 네트워크 트래픽 서비스 분석 방법연구에 대하여 소개를 한다. 3장에서는 SSL/TLS 프로토콜에 대한 소개와 4장에서는 본 논문에서 제안하는 SSL/TLS 서비스 분석 방법에 대한 설명을 한다. 5장에서는 4장을 바탕으로 구축한 시스템을 이용하여 실제 트래픽 분류 실험 및 결과를 기술한다. 마지막으로 6장에서는 결론 및 향후 연구계획에 대해서 기술한다.

Ⅱ. 관련 연구

네트워크 트래픽의 서비스를 분류하는 방법은 다양한 방향으로 연구가 발전되어 오고 있다. 먼저 헤더 시그니쳐 기반 분류 방법[4]은 패킷의 헤더정보를 기반으로 서비스를 식별하는 가장 기초적인 방법으로, Firewall은 포트정보에 기반하여 서비스를 식별하고 제어하는 대표적인 예 이다. 하지만 현재 인터넷 트래픽의 상당수를 차지하고 있는 P2P 프로그램을 비롯하여 많은 응용 프로그램들은 포트정보를 공개하지 않거나, 동적 포트를 사용하기 때문에 분석률과 정확도가 떨어져 현재의 트래픽 분류에는 부적합 하다.

다음으로 페이로드상의 문자열에 직접적으로 시그니쳐를 매칭 시켜 서비스를 분류해내는 페이로드 시그니쳐 기반 분류 방법이 있다. 이 방법은 패킷의 페이로드를 직접 검사를 하기 때문에 높은 정확도와 분석률을 보장하는 반면, 다른 방법에 비해 상대적으로 높은 부하를 발생시키며 처리 속도가 느린 단점을 갖는다. 이러한 단점을 극복하기 위해 다양한 연구[5]가 진행되었지만, 여전히 시그니쳐를 수작업으로 찾아야 하기 때문에 그 과정에서 많은 시간과 노력이 소요된다. 또한 인터넷 기반 응용 서비스의 라이프 사이클이 짧기 때문에 시그니쳐의 갱신 작업이 빈번하게 요구되며, 암호화된 데이터의 경우 시그니쳐를 생성하는데 어려움이 있다.

마지막으로 패킷의 크기, 윈도우 크기, 수집시간 등의 플로우의 통계 정보 기반으로 서비스를 분류하는 통계정보 시그니쳐 기반 분류 방법[6]이 있다. 통계정보 시그니쳐 기반 분류 방법은 페이로드를 직접 검사하지 않고 플로우의 통계정보를 이용하여 트래픽을 분석하기 때문에 빠른 분석속도와 암호화 되어 있어도 분류가 가능하다는 장점이 있다. 이러한 특성을 이용하여 암호화 프로토콜인 SSL/TLS트래픽을 분류하는 연구[7]가 진행이 되었다. 평균 패킷 크기, Inter-arrival time 등 22가지의 통계 정보를 이용하여 “AdaBoost”, “C4.5”, “Naïve Bayes”, 등 Machine Learning 기반 알고리즘으로 SSL/TLS 트래픽을 식별하였다. 하지만 Machine Learning 알고리즘의 단점인 특정 네트워크에 종속적이라는 것과 서비스별로 세부적인 분류를 하지 못하는 한계점을 벗어나지는 못하였다. 본 논문에서 제안하는 방법은 프로토콜 단위 또는 암호화 여부에 대한 분석에 초점이 맞춰진 기존 SSL/TLS 트래픽 분류방법의 한계를 극복하고, SSL/TLS 트래픽을 응용 서비스 단위로 분석하는데 차별성이 있으며, 이는 본 연구 분야에서는 새로운 시도이다.

Ⅲ. SSL/TLS

SSL/TLS 프로토콜은 1994년 넷스케이프사에서 개발된 웹 서버간의 안전한 데이터 전송을 위해 데이터를 평문이 아닌 암호화하여 전송하는 기술로 본 장에서는 SSL/TLS 프로토콜의 구조와 각 필드별 기능, 그리고 SSL/TLS Handshake과정에 대하여 기술한다.

3.1 SSL/TLS Protocol

그림 1은 SSL/TLS 프로토콜의 구조를 나타낸 것이다. SSL/TLS 프로토콜은 상위 3개의 프로토콜과 가장 하위에 레코드 프로토콜로 이루어져 있으며 TCP

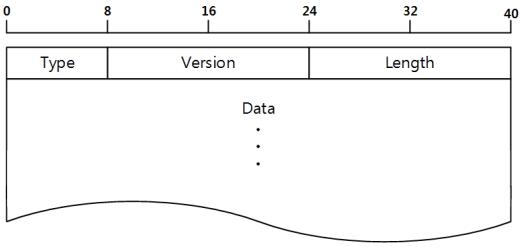

그림 1. SSL/TLS 프로토콜의 구조
Fig. 1. SSL/TLS Protocol's Structure

가 데이터를 패킷 단위로 나누어 사용하듯, SSL/TLS는 데이터를 레코드단위로 나누어 사용한다. SSL/TLS 레코드 프로토콜은 계층적 구조로써, 각각의 계층에서 메시지는 레코드의 종류, SSL/TLS 버전, 데이터의 길이, 데이터를 위한 필드를 갖는다.

표 1은 Type필드로 1바이트로 이루어져 있으며, 레코드의 프로토콜 종류를 명시한다. Change cipher spec은 Handshake에 의해 협상된 압축, MAC, 암호화 방식 등이 이후부터 적용됨을 상대방에게 알려준다. Alert 은 SSL/TLS 관련 경고 메시지를 압축, 암호화하여 전달한다. 또한 Handshake는 서버와 클라이언트간의 상호인증을 수행하고, 사용할 키 교환 방식, 대칭키 암호 방식 등의 보안 속성을 협상한다. 마지막으로 Application data는 실제 암호화된 데이터를 전송하는 프로토콜에 해당한다.

Version 필드는 2바이트로 이루어져 있으며, 사용하는 SSL/TLS의 버전을 나타낸다. SSL 버전 1은 실제로 발표되지는 않았고, SSL 버전 2가 이후 공개적으로 발표가 되었는데 많은 보안 취약점이 있었다. 그것을 보완한 것이 SSL 버전 3이며 공식적으로 표준화한 이름이 TLS 이다. 이후로 버전이 올라감에 따라 0.1씩 더하여 버전 넘버링을 하였다.

표 1. 프로토콜 종류
Table 1. Protocol Type

code	Protocol Type
20	Change cipher spec
21	Alert
22	Handshake
23	Application data

표 2. SSL/TLS 버전
Table 2. SSL/TLS Version

code	SSL/TLS Version
0300	SSL 3.0
0301	SSL 3.1 (SSL/TLS 1.0)
0302	SSL 3.2 (SSL/TLS 1.1)
0303	SSL 3.3 (SSL/TLS 1.2)

다음은 Length필드로 뒤이어 나오는 데이터의 길이를 Big-Endian방식으로 2바이트를 사용하여 표기한다. 예를 들어 L1L2라는 길이가 표기 되어 있으면, 256·L1+ L2가 데이터의 길이가 된다.

마지막으로 Data필드는 실제 서버와 클라이언트가 필요로 하는 정보를 전달하는 내용으로 암호화된 데이터와 무결성 검사용 데이터들이 최대 18KB까지 발생한다.

3.2 SSL/TLS Handshake 프로토콜

그림 2는 SSL/TLS Handshake 과정을 도식화 한 것으로 먼저 Client는 Server에게 ClientHello 메시지를 보내 연결을 시도함과 동시에 자신이 사용 가능한 CipherSuite를 Server에게 전송한다.

Client로부터 Hello메시지를 받은 Server는 ServerHello, Certificate, ServerHelloDone 메시지를 Client측으로 응답한다. Server Hello는 ClientHello 메시지를 수신했다는 응답이며, 이때 세션ID가 생성되어 전송된다. Certificate 메시지는 Public key를 담고 있는 Server의 인증서를 보낸다. 이때 Client가 인

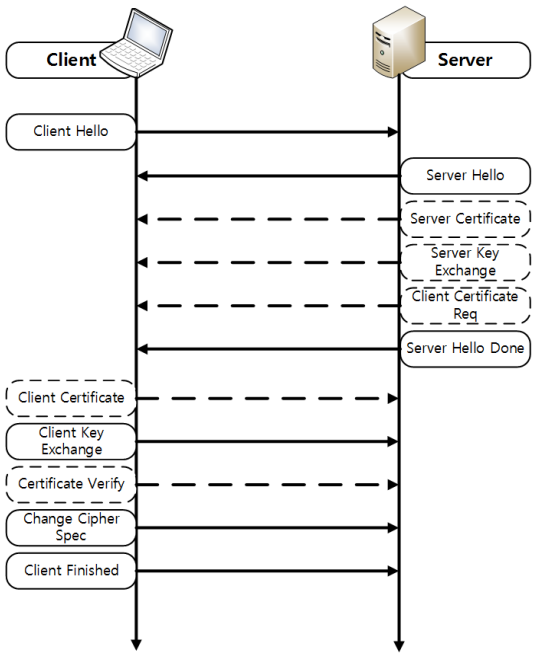

그림 2. SSL/TLS Handshake 프로토콜
Fig. 2. SSL/TLS Handshake Protocol

증서를 보내지 말라고 요청을 할 때에만 인증서를 보내지 않으며, 그 외에는 모두 보낸다.

ServerKeyExchange 메시지는 키 교환시 인증서만으로 충분하지 않은 경우 사용되는 메시지이다.

CertificateRequest는 Client의 인증서를 요청하는 메시지로 Server가 추가 인증을 필요로 하는 경우 요청하는 메시지 이다.

마지막으로 ServerHelloDone 메시지는 ServerHello가 끝났으며, Client가 응답을 시작해야 함을 알린다.

ServerHelloDone 메시지를 받은 Client는 Server로부터 인증서요청을 받았다면 Certificate메시지를 통해 Client 인증서를 전송한다. 이후 ClientKeyExchange 메시지를 통해 키 교환을 하고, CertificateVerify메시지를 통해 Client의 인증서가 신뢰 할 수 있음을 알린다.

Ⅳ. SSL/TLS Service Identification Method

본 장에서는 본 논문에서 제안하는 SSL/TLS 프로토콜을 사용하는 서비스 식별 방법(SSIM : SSL/TLS Service Identification Method)에 대하여 기술하며, 분석하고자 하는 서비스를 네트워크 서비스를 제공하는 서버라 정의한다.

SSL/TLS 프로토콜은 TCP와 마찬가지로 Handshake과정을 거쳐 세션을 맺는 연결지향 프로토콜이다. 암호화된 페이로드를 이용하여 서비스를 식별하는 데에는 어려움이 있지만, 암호화 알고리즘, 인증서와 키를 교환하는 단계는 암호화가 되지 않은 평문으로 교환하기 때문에 SSL/TLS Handshake과정의 데이터를 분석하여 서비스를 식별 할 수 있다.

그림 3은 제안하는 방법의 전체 구성도를 나타낸다. SSIM의 입력은 트래픽 수집기에서 수집한 트래픽을 5-Tuple(Source IP, Destination IP, Source Port, Destination Port, Protocol)정보 기반으로 다양한 통계정보와 최대 20번째 패킷까지의 페이로드를 저장한 플로우 파일이다.

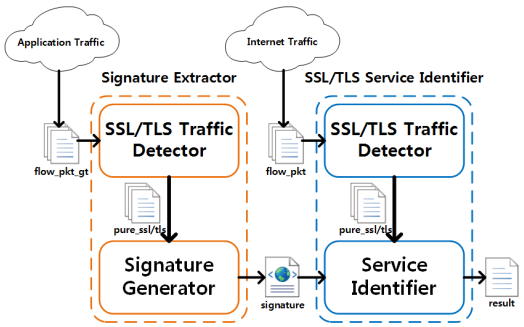

그림 3. SSL/TLS Service Identification Method
Fig. 3. SSL/TLS Service Identification Method

본 방법은 크게 두 부분으로 나뉘는데, 먼저 “Signature Extractor”는 시그니쳐를 추출하고자 하는 타깃 응용 서비스의 트래픽만을 수집한 트래픽 파일을 입력으로 받아 해당 서비스의 페이로드 시그니쳐를 추출한다.

다음으로 “SSL/TLS Service Identifier”는 “Signature Extractor”에서 추출한 시그니쳐와 SSL/TLS 필드 정보를 이용하여 실제 온/오프라인 트래픽을 응용 서비스 별로 식별한다. 각 모듈에 대한 자세한 내용은 뒤이어 한다.

4.1 Signature Extractor

“Signature Extractor”는 플로우 파일을 분석하여 SSL/TLS 서비스 분석에 필요한 페이로드 시그니쳐를 자동으로 생성하는 역할을 수행한다.

“Signature Extractor”는 두 가지 모듈로 나뉘는데, “SSL/TLS Traffic Detector”는 입력으로 받은 플로우 파일 중 순수한 SSL/TLS 플로우만을 식별하여 다음 단계로 전달한다. 이는 SSL/TLS 응용을 식별하는 본 방법의 특성상 SSL/TLS 이외의 프로토콜을 사용하는 트래픽을 제외시켜 추출되는 시그니쳐의 정확도를 높이기 위한 과정이다.

“SSL/TLS Traffic Detector”는 플로우 파일을 입력으로 받아 각 플로우 별 첫번째 패킷을 검사한다. SSL/TLS의 레코드는 TCP/IP보다 상위 단에 있기 때문에 페이로드의 일부로 인지하여 저장한다. 따라서 페이로드에 기록된 레코드의 헤더 정보만을 이용하여 SSL/TLS 임을 식별할 수 있다. SSL/TLS 레코드의 처음 5바이트는 레코드 헤더로 해당 레코드의 종류, 버전, 길이 정보가 기록 되어 있다. 이 헤더가 사전에 정의해 놓은 헤더의 범위와 일치하고, 길이정보와

표 3. SSL/TLS Traffic Detector
Table 3. SSL/TLS Traffic Detector

TYPE = 20 | 21 | 22 | 23
VERSION = 0300 | 0301 | 0302 | 0303
LENGTH = Record + Record header length

```
1:  SSL/TLS Traffic Detector(Flown) {
2:    if ( First packet of Flow )
3:      if ( !TYPE ) return NO
4:      if ( !VERSION ) return NO
5:      if ( LENGTH != payload_len ) return NO
6:      return YES
7:  }
```

Data 필드의 길이가 일치하면 해당 플로우를 SSL/TLS 트래픽이라 식별 할 수 있다.

"SSL/TLS Detector"에 의해 식별된 SSL/TLS 플로우는 "Signature Generator"에 입력되어 인증서를 교환하는 Certificate 레코드의 CommonName필드를 시그니쳐로 추출한다.

표 4는 "Signature Generator"의 알고리즘이다. SSL/TLS는 레코드단위로 데이터를 분리하지만 실제 트래픽상에서는 패킷으로 구분되어 이동하기 때문에 방향이 같은 레코드가 연속적으로 발생하였을 때는 다수의 레코드가 하나, 또는 다수의 패킷에 걸쳐서 전송된다. 따라서 SSL/TLS Handshake 과정에서 ServerHello와 Certificate는 연속적으로 발생하므로 하나, 또는 다수의 패킷에 연속되어 기록된다. 본 방법에서는 Certificate의 CommonName필드를 시그니쳐로 추출하지만 실제로는 Certificate가 독립적인 패킷으로 전송이 되지 않기 때문에 ServerHello로 시작되는 패킷을 검사하게 된다. 해당 패킷에서 CommonName의 ID를 탐색하여 뒤이어 기록되어 있는 길이 정보를 토대로 CommonName을 시그니쳐로 추출한다. 이때 CommonName은 인증서 발행인인 Issuer와 그 대상인 Subject로 구분이 되는데 해당 필드의 ID가 같기 때문에 기록 순서로 구분할 수 있다. 그 순서는 Issuer필드 이후에 Subject필드가 명시되어 있는데, 이중 시그니쳐로써 가치가 있는 필드는 Subject의 CommonName이므로 두번째로 명시되어 있는 CommonName을 시그니쳐로 선정한다. "Signature Extractor"에서 출력된 시그니쳐의 집합은 "Service Identifier"의 입력으로 사용된다.

표 4. Signature Generator
Table 4. Signature Generator

```
COMMONNAME_ID = 0x55x04x03          (CM_ID)
COMMONNAME = The string after CM_ID
 1:  Signature Generator( SSL/TLS Flown ) {
 2:    convert packets into SSL/TLS record sequence
 3:    if ( 2nd record != SERVER_HELLO )
 4:      return Sig. NOT found
 5:    if ( 3rd record == CERTIFICATE ) {
 6:      find subject's COMMONNAME_ID
 7:      return COMMONNAME as a Sig.
 8:    }
 9:    else return Sig NOT found
10:  }
```

4.2 SSL/TLS Service Identifier

"SSL/TLS Service Identifier"는 분석할 트래픽의 플로우 파일과 분석하는데 사용할 "Signature Extractor"에서 생성한 시그니쳐의 집합을 입력 받아 인터넷 트래픽의 서비스를 식별한다. "SSL/TLS Service Identifier"의 구조는 "Signature Extractor"와 마찬가지로 두 가지 모듈로 나뉘며 시스템의 흐름 역시 "Signature Extractor"와 비슷하다.

"SSL/TLS Traffic Detector"는 "Signature Extractor"에서와 마찬가지로 입력으로 받은 플로우 파일 중 순수한 SSL/TLS 플로우만을 식별하여 다음 단계로 전달하는데, "Signature Extractor"에서는 SSL/TLS 이외의 프로토콜을 사용하는 트래픽을 제외시켜 시그니쳐의 정확도를 높이기 위함이었던 반면, "SSL/TLS Service Identifier"에서는 분석 속도를 향상시키기 위한 것이 목적인 차이가 있다. 알고리즘 역시 동일하기 때문에 설명은 생략한다.

"Service Identifier"는 "Signature Extractor"에서 생성된 시그니쳐를 입력받아 "SSL/TLS Traffic Detector"에서 식별된 SSL/TLS 플로우를 서비스 별로 분석하는 모듈로 전체적인 형태는 시그니쳐 추출 단계의 "Signature Generator"와 흡사하다. 다만 다른 점은 시그니쳐를 추출하여 기록하는 단계가 시그니쳐를 매칭시키는 부분으로 바뀌었다. SSL/TLS 서비스의 시그니쳐는 특정 위치의 필드에서 추출 한 것이므로 반대로 분석을 할 때에도 같은 필드에 시그니쳐를 매칭 시키면 빠르고, 정확하게 서비스를 분석할 수 있기 때문이다.

표 5. Service Identifier
Table 5. Service Identifier

```
COMMONNAME_ID = 0x55x04x03          (CM_ID)
COMMONNAME = The string after CM_ID
 1:  Service Identifier ( SSL/TLS Flown, Sign ) {
 2:    convert packets into SSL/TLS record sequence
 3:    if ( 2nd record != SERVER_HELLO )
 4:      return The flow is unidentified service
 5:    if ( 3rd record == CERTIFICATE ) {
 6:      find subject's COMMONNAME_ID
 7:      if ( COMMONNAME == Sign )
 8:        The flow is identified as Sign service
 9:    }
10:    else return The flow is unidentified service
11:  }
```

V. 실험 및 성능 평가

본 장에서는 학내에서 발생하는 트래픽과 본 논문 저자의 Local PC에서 수집한 트래픽을 분석하여

SSL/TLS트래픽의 사용 빈도, 포트 분포, 각 서비스별 생성된 시그니쳐 개수와 분석률을 구하는 실험과 그 결과를 기술한다. 명확한 실험 결과를 위하여 실험 데이터는 전처리로 TCP 이외의 프로토콜을 사용하는 트래픽과 수집 시간에 의해 플로우가 처음부터 기록되지 않은 플로우를 제외 하였다. 또한 Handshake가 제대로 이루어지지 않아 실제 전송된 데이터가 없는 플로우 정보 역시 제거하여 명확한 TCP 데이터만을 이용하여 실험에 사용 하였다. 본 논문에서 수행한 SSL/TLS 응용 트래픽의 페이로드 시그니쳐를 자동으로 생성하고 이를 검증하는 실험은 처음으로 수행된 연구로 실험의 결과는 비교대상이 없다.

표 6은 SSL/TLS로 암호화 되는 프로토콜의 종류와 각 프로토콜의 포트번호 이다. 이외에도 많은 SSL/TLS를 사용하는 포트번호가 있지만 본 논문에서는 기존 TCP 프로토콜을 SSL/TLS로 암호화 한 프로토콜의 포트번호를 Well Known Port라 정의한다.

SSL/TLS 트래픽의 사용량을 분석하기 위하여 학내에서 하루 동안 발생한 트래픽을 수집하여 “SSL/TLS Traffic Detector”을 이용해 SSL/TLS 트래픽 사용량을 측정하였다. 표 7은 실험의 결과로 세 가지 전처리 과정을 거친 전체 트래픽은 플로우단위로 약 $6 \cdot 10^6$이며, 이 중 SSL/TLS 트래픽의 사용 빈도는 8.38%이다. 이를 Well known 포트를 사용하는 SSL/TLS 트래픽과 사용하지 않는 트래픽으로 나누면 표 8과 같다.

SSL/TLS 트래픽으로 분류된 데이터 중 대부분은 Well Known Port를 사용하지만, 이 외의 포트를 사용 하는 트래픽은 서비스 제공자와 클라이언트 간의 사전 계약을 통하여 서비스 제공자가 임의로 지정한 포트를 사용하여 데이터를 교환한다. 또한, 기존 TCP 포트를 사용하지만 Stunnel등의 프로그램을 이용하여 터널링된 경우에 Well Known Port 이외의 포트를 사용한다. 반면, SSL/TLS가 아닌 트래픽 중 Well Known Port를 사용하는 트래픽이 소량 발생 하는데 이는 SSL/TLS Handshake가 제대로 이루어지지 않아 세션이 맺어지지 않거나, 의미 없는 데이터들로 채워져 있는 경우이다.

표 9는 SSL/TLS 트래픽으로 식별된 트래픽을 포트 별 분포도를 나타낸 것이다. 443(HTTPS) 포트를 사용하는 트래픽이 대부분을 차지하는데 이는 전체 SSL/TLS 트래픽 중 약 99%를 차지 하는 정도 이다.

표 6. SSL/TLS을 사용하는 Port
Table 6. Well Known Port using SSL/TLS

Protocol	port
HTTPS	443, 8531
IMAPS	993
POP3S	995
SMTPS	465, 25, 587, 2526
FTPS	990, 989
TELNETS	992
IRCS	994, 6679, 6697
SIPS	5061
LDAPS	636, 3269
NNTPS	563
MMS-SSL	695
OPCS	4843
XMPPS	5223
OFTPS	6619
MQQTS	8883

표 7. SSL/TLS 분석
Table 7. SSL/TLS Analysis

	Flows	Packets	Bytes
Total	6,043,560	1,080,739,271	930,080,255,327
SSL/TLS	506,412	59,164,220	46,224,581,249
Rate	8.38%	5.47%	4.97%

표 8. SSL/TLS를 사용하는 포트 분포
Table 8. SSL/TLS Analysis with Well Known Port

Port		SSL/TLS	Non SSL/TLS
SSL/TLS Port	Flow	503·103	27·103
	Rate	8.32 %	0.44 %
Non SSL/TLS Port	Flow	3·103	5,510·103
	Rate	0.06 %	91.18 %
Total	Flow	506·103	5,537·103
	Rate	8.38 %	91.62 %

표 9. SSL/TLS Port 분포
Table 9. SSL/TLS Port Distribution

Port	Flow	Packet	Byte
Total	506,412	59,164,220	46,224,581,249
SSL/TLS Port	502,898	58,473,474	45,836,333,614
	99.31%	98.83%	99.16%
25(SMTPS)	3	2,334	199,592
443(HTTPS)	499,265	55,979,262	43,468,435,703
465(SMTPS)	3	7,888	7,582,259
587(SMTPS	1	55	15,215
993(IMAPS)	3,115	2,418,896	2,335,252,130
994(IRCS)	67	17,070	7,219,156
995(POP3S)	181	13,458	9,176,575
5223(XMPPS)	263	34,511	8,452,984
Non SSL/TLS Port	3,514	690,746	338,247,635
	0.69%	1.17%	0.84%

따라서 서비스 분석 실험에서 443 포트를 사용하는 서비스만을 선정하여 분석 실험을 진행 하였다. 반대로 SSL/TLS가 아닌 트래픽 중 SSL/TLS라고 알려진 포트를 사용한 트래픽의 비율은 표 10과 같다.

SSL/TLS 트래픽이 아니라 식별한 트래픽 중 SSL/TLS로 알려진 포트를 사용하는 트래픽은 극소량 발생 하는데 이 중 대부분은 SSL/TLS Handshake를 명확히 맺지 못하거나, 레코드의 종류가 Alert과 같이 Server와 Client간 실제 의미 있는 데이터를 주고 받지 않는 경우이다.

다음으로 추출한 시그니처의 정확도를 알아보기 위한 실험을 하였다. 실험 방법은 본 논문 저자의 Local PC에서 트래픽을 서비스별로 다양한 기능들을 수행하는 동시에 수집하여 "Signature Generator"를 이용하여 시그니처를 추출하였다. 이후 다시 한번 서비스별로 트래픽을 수집해 "SSL/TLS Service Identifier"를 이용하여 분석하였다. Target 서비스는 SSL/TLS 트래픽의 약 99%를 차지 하고 있는 443 포트를 사용하는 트래픽 중 최근 사용량이 늘어나고 있는 "Google", "Kakaotalk", "Facebook" 3가지 서비스를 선정 하였다.

표 11은 각 Target 서비스 별 추출한 시그니처의 개수이다. 시그니처는 트래픽을 장시간 수집할 수록 많은 양이 추출된다. 실험에 사용된 시그니처는 약 10분간 수집한 트래픽을 기반으로 생성된 시그니처이다. 초기 추출된 시그니처는 순수한 Target 서비스의 시그니처가 아닌 경우에 본 저자가 제외를 하였다. facebook서비스를 사용하는데 배너 형식으로 발생하는 Google광고와 Youtube 동영상 등이 그 예시 이다.

표 10. Non-SSL/TLS Port 분포
Table 10. Non-SSL/TLS Port Distribution

Port	Flow	Packet	Byte
Total	5,537,148	1,021,575,051	883,855,674,078
SSL/TLS Port	26,661	1,576,968	777,435,396
	0.48%	0.15%	0.09%
25(SMTPS)	117	65,995	65,159,564
443(HTTPS)	15,384	931,226	431,628,524
993(IMAPS)	760	203,435	188,514,544
995(POP3S)	352	26,250	12,317,569
5223(XMPPS)	10,047	350,051	79,814,241
6697(IRCS)	1	11	954
Non SSL/TLS Port	5,510,487	1,019,998,083	883,078,238,682
	99.52%	99.85%	99.91%

표 11. 시그니처
Table 11. Sigantures

Service		Signature
google	google	*.google.com
		www.google.com
		*.google.co.kr
	youtube	*.doubleclick.net
		*.g.doubleclick.net
	gmail	*.mail.google.com
		mail.google.com
	login	accounts.google.com
	googledocs	*.c.docs.google.com
	blogger	*.blogger.com
	etc	www.googleadservices.com
		*.google-analytics.com
		developer.android.com
		*.googleusercontent.com
		checkout.google.com
		*.googleapis.com
facebook		*.facebook.com
		*.atlassolutions.com
kakaotalk		*.kakao.com
		*.talk.kakao.com

또한 CDN[8]으로 서비스를 제공하는 경우의 시그니처도 추출이 되었는데 이 경우 역시 특정 서비스의 시그니처라고 할 수 없기 때문에 제외 하였다.

표 12는 각 서비스 별로 추출한 시그니처를 이용하여 트래픽을 분석한 결과이다. 독립된 응용프로그램을 사용하는 Kakaotalk에 비하여 Explorer, Chrome등 브라우저를 이용하는 Web서비스의 일종인 Google과 Facebook은 분석률이 낮은데, 이는 Web서비스의 특성상 하나의 Web page에서 다양한 서버에서 서비스들이 발생하기 때문에 순수한 서비스의 트래픽을 모으기 힘들기 때문이다. 실제로 "SSL/TLS Traffic Detector"를 이용하여 순수한 SSL/TLS 트래픽을 추출하기 전에는 Http 트래픽등 불순물이 다량 섞여있었다. 또한, 서버가 세션ID를 기억하는 경우에는 일반적인 Handshake가 아닌 인증서와 키 교환 등의 과정이 생략된 Abbreviated Handshake를 통해 세션을 맺기 때문에 Certificated를 검사하여 분석하는 본 방법만으로는 분석하기 힘들다. 따라서 이와 같은 경우에는 추가적인 분석 방법이 결합되어야 한다.

표 12. 서비스별 분석 정확도
Table 12. Accuracy of Service Classification

Service		Flow	Packet	Byte
Google	Total	689	68,755	54,252,683
	Classified	559	62,881	50,022,915
	Accuracy	81.13%	91.46%	92.20%
Facebook	Total	227	68,216	66,362,942
	Classified	155	60,071	60,520,703
	Accuracy	68.28%	88.06%	91.20%
Kakaotalk	Total	112	6,020	4,584,514
	Classified	104	5,646	4,421,168
	Accuracy	92.86%	93.79%	96.44%

Ⅵ. 결 론

본 논문에서는 SSL/TLS 프로토콜을 이용하는 서비스의 시그니쳐를 SSL/TLS Handshake 과정에서 자동으로 추출하고, 그 시그니쳐를 이용하여 서비스를 식별하는 방법을 제안하였으며, 이를 실험을 통해 성능을 검증하였다.

그러나 하나의 Web page에서 다양한 서비스를 제공하는 Web 서비스의 분석률이 기대에 미치지 못하였으며, Abbreviated Handshake를 통하여 세션을 맺는 경우 Certificate가 생략되기 때문에 이를 분석하지 못하는 문제점이 있었다. 따라서 향후 앞서 제시한 문제점을 보완 하여 SSIM의 분석률과 정확도를 개선시키고자 한다.

References

[1] RFC 5246, *The Transport Layer Security (TLS)* Protocol Version 1.2, Retrieved 16, Feb. 2015, https://tools.ietf.org/html/rfc5246

[2] K.-L. Kim, M.-S. Kim, and H. Kim, “SSH traffic identification using EM clustering,” *J. KICS*, vol. 37, no. 12, pp. 1160-1167, 2012.

[3] J.-S. Park, S.-H. Yoon, Y. Won, and M.-S. Kim, “A lightweight software model for signature-based application-level traffic classification system,” *IEICE Trans. Inf. Syst.*, vol. 97, no. 10, pp. 2697-2705, 2014.

[4] S.-H. Yoon, J.-S. Park, and M.-S. Kim, “Header signature maintenance for internet traffic identification,” *KNOM Rev.*, vol. 16, no. 1, Jul. 2013.

[5] J.-S. Park, S.-H. Yoon, and M.-S. Kim, “Performance improvement of the payload signature based traffic classification system using application traffic locality,” *J. KICS*, vol. 38, no. 7, pp. 519-525, 2013.

[6] H.-M. An, J.-H. Ham, and M.-S. Kim, “Performance improvement of the statistical information based traffic identification system,” *KIPS Trans. Computer and Commun. Syst.(KTCCS)*, vol. 2, no. 8, pp. 335-342, Aug. 2013.

[7] C. McCarthy and A. N. Zincir-Heywood, “An investigation on identifying SSL traffic,” *2011 IEEE Symp. CISDA*, pp. 115-122, Paris, France, Apr. 2011.

[8] S.-H. Kong and J.-Y. Lee, “Effective contents delivery system using service adaptive network architecture(SaNA),” *J. KICS*, vol. 39, no. 6, pp. 406-413, 2014.

김 성 민 (Sung-Min Kim)

2014년 : 고려대학교 컴퓨터정보학과 학사
2014년~현재 : 고려대학교 컴퓨터정보학과 석사과정
<관심분야> 네트워크 관리 및 보안, 트래픽 분석, 데이터 암호화

박 준 상 (Jun-Sang Park)

2008년 : 고려대학교 컴퓨터정보학과 학사
2011년 : 고려대학교 컴퓨터정보학과 석사
2014년 : 고려대학교 컴퓨터정보학과 박사
<관심분야> 네트워크 관리 및 보안, 트래픽 분석

윤 성 호 (Sung-Ho Yoon)

2009년 : 고려대학교 컴퓨터정보학과 학사
2011년 : 고려대학교 컴퓨터정보학과 석사
2015년 : 고려대학교 컴퓨터정보학과 박사

<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석

김 종 현 (Jong-Hyun Kim)

1995년~1998년 : 삼성전자 SW 연구개발 연구원
2000년 : 오클라호마 주립대학교 컴퓨터과학과 공학석사
2005년 : 오클라호마 주립대학교 컴퓨터과학과 공학박사
2005년~현재 : 한국전자통신연구원 책임연구원

<관심분야> 정보보호, 네트워크보안, 네트워크 포렌식

최 선 오 (Sun-Oh Choi)

2005년 : 고려대학교 컴퓨터학과 학사
2008년 : 고려대학교 컴퓨터학과 석사
2014년 : Purdue 대학교 전자 및 컴퓨터공학과 박사
2014년~현재 : 한국전자통신연구원 선임연구원

<관심분야> 네트워크보안, 데이터 보안

김 명 섭 (Myung-Sup Kim)

1998년 : 포항공과대학교 전자계산학과 학사
2000년 : 포항공과대학교 컴퓨터공학과 석사
2004년 : 포항공과대학교 컴퓨터공학과 박사
2006년 : Dept. of ECS, Univ. of Toronto, Canada
2006년~현재 : 고려대학교 컴퓨터정보학과 부교수

<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석, 멀티미디어 네트워크