논문 12-37B-12-08

한국통신학회논문지 '12-12 Vol.37B No.12
http://dx.doi.org/10.7840/kics.2012.37B.12.1160

EM 클러스터링을 이용한 SSH 트래픽 식별

김 경 륜◆, 김 명 섭*, 김 형 중°

SSH Traffic Identification Using EM Clustering

Kyoung-lyoon Kim◆, Myung-sup Kim*, Hyoung-joong Kim°

요 약

네트워크 트래픽 모니터링에 있어서 트래픽을 사용하는 목적을 알아내는 것은 서비스 품질, 방화벽의 동작, 보안 측면에 있어서 중요한 이슈가 되고 있다. 트래픽을 사용하는 목적을 알게 되면 이를 방화벽에서 거부하거나 허용할 수 있고 이는 서비스 품질, 보안적 측면에서 효과적인 운용이 가능해진다. 하지만 수많은 어플리케이션은 보안이나 서비스 측면에서 트래픽을 암호화시키고 있어 효과적인 트래픽 모니터링이 어렵다. 본 논문에서는 암호화된 트래픽을 사용하는 SSH(Secure Shell) 프로토콜을 분석하고 SSH 터널링, SFTP(SSH File Transfer Protocol)와 일반 SSH 트래픽의 차이점을 분석하고 식별할 수 있는 방법을 제시하고 실험을 통해 검증했다.

Key Words : Traffic Monitoring, Encrypted Traffic, Network Security, IDS, EM Clustering, 패킷분석

ABSTRACT

Identifying traffic is an important issue for many networking applications including quality of service, firewall enforcement, and network security. Once we know the purpose of using the traffic in the firewall, we can allow or deny it and provide quality of service, and effective operation in terms of security. However, a number of applications encrypts traffics in order to enhance security or privacy. As a result, effective traffic monitoring is getting more difficult. In this paper, we analyse SSH encrypted traffic and identify differences among SSH tunneling, SFTP, and normal SSH traffics. By using EM clustering, we identify traffics and validate experiment results.

Ⅰ. 서 론

최근 인터넷 사용자의 증가와 초고속 네트워크의 보급으로 네트워크 트래픽이 급증하고 있다. 트래픽의 어플리케이션 분석은 대역폭 관리, 서비스 품질 관리 등을 위한 효과적인 네트워크 운용, 방화벽의 성능향상, 보안성능 향상 등에 필요하다. 그런데 초고속 네트워크는 웹 서비스, e-mail, ftp, telnet 같은 단순한 서비스에서 P2P, 메신저, 온라인 게임 등 다양한 트래픽을 발생시키는 상황에 이르렀다. 그 중에 SSH(Secure Shell) 프로토콜을 사용하는 터널링 기법은 네트워크상에서 금지된 서비스를 사용할 수 있는 것이 가능하기 때문에 이를 탐지하는 방법이 필요성이 커지고 있다.

그동안 트래픽 식별의 고전적인 방법으로는 IANA[1]에 포트를 등록한 어플리케이션을 탐지하는 방법이 있다[2,4]. 하지만 이러한 방법은 등록된 포트를 사용하지 않는 어플리케이션의 증가와 동적 포트를 사용하는 어플리케이션으로 인해 분류가 힘들어지고 있다. 다른 방법으로는 패킷의 페이로드를

※ 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(2012R1A2A2A01015587)
◆ 주저자 : 고려대학교 정보보호대학원 멀티미디어보안연구실, kklyoon@korea.ac.kr, 준회원
° 교신저자 : 고려대학교 정보보호대학원 멀티미디어보안연구실, khj-@korea.ac.kr, 종신회원
* 고려대학교 과학기술대학 컴퓨터정보학과 네트워크관리연구실, tmskim@korea.ac.kr, 종신회원
논문번호 : KICS2012-10-493, 접수일자 : 2012년 10월 14일, 최종논문접수일자 : 2012년 11월 23일

검사하는 방법[2,3]이 있는데 이러한 방법은 암호화된 트래픽에서는 사용이 불가능하고 심층패킷분석에 따른 법적인 문제가 있어 실재 환경에서 적용이 힘들다.

본 논문에서는 암호화된 SSH를 각기 다른 목적으로 사용하는 트래픽을 수집하여 정상사용 트래픽, 터널링 트래픽, SFTP(SSH File Transfer Protocol) 트래픽으로 구분할 수 있는 기준을 세우고 이를 클러스터링했다.

본 논문은 다음과 같은 순서로 구성되어 있다. 2장에서는 관련연구를 기술하고, 3장에서는 SSH 프로토콜과 터널링 기술, SFTP에 관한 개요를 기술한다. 4장에서는 각 트래픽의 특징을 그래프와 함께 분석하고, 5장에서는 데이터 수집과정과 실험방법을 기술한다. 6장에서는 제안된 방법으로 실제 실험해 결과를 보여주며, 마지막으로 7장에서는 결론 및 향후 연구 과제를 기술한다.

Ⅱ. 관련 연구

암호화된 트래픽에 대한 분류는 기존의 많은 연구들에서 응용 트래픽 분류를 위한 DPI(deep packet inspection) 기반, 기계학습 기반, 패킷 사이즈와 포트 기반 방식들이 제안되고 있지만 분류의 정확성이 아직은 크게 높지 않아 향후 연구로 남겨두고 있다.

한 선행연구[11]에서는 SSH 프로토콜에 대해서 처음 연결을 맺는 과정에서 발생되는 패킷의 64바이트를 검사하여 이를 AdaBoost, hidden Markov, naive Bayesian, maximum entropy 모델을 사용해 분류했다. 탐지율은 86%, 오탐율은 0%의 결과를 얻었다.

좀 더 많은 응용프로그램을 사용한 연구[12]에서는 웹, e-mail, P2P, VoIP 등 다양한 트래픽을 수집하여 플로우의 처음 5개의 패킷 정보를 가지고 C4.5 알고리즘을 사용해 실시간 탐지실험에서 좋은 결과를 얻었다. 하지만 별도의 학습을 위한 데이터 세트가 필요하고 데이터 세트과 차이점이 많은 SSH와 같은 대화형 프로토콜에 대해서는 30% 이하의 낮은 탐지율을 보이는 한계점을 보였다.

연구[13]에서는 SSL의 처음 연결에 사용된 4개 패킷의 5가지 항목(목적지/발신지 IP 주소, 목적지/발신지 포트 번호, 프로토콜)과 패킷 사이즈를 사용하여 탐지하는 방법을 제시했다. 그러나 이 방법은 동적 포트 할당을 사용하는 프로토콜에 대해서는 탐지를 하지 못하는 문제점이 있다.

연구[14,15]에서는 패킷 사이즈, 시간, 패킷의 방향 등을 사용하여 k-nearest neighbor 방법과 hidden Markov model 학습 시스템을 실험을 통해 비교했다. 그 결과 SSH 트래픽 탐지율은 76%, 오탐율은 8%의 결과를 냈다.

연구[16]에서는 플로우 당 평균 패킷 사이즈와 inter-arrival time을 가지고 비선형 회귀분석법으로 SSH 트래픽을 89% 비율로 탐지하는 결과를 얻었다.

일반적인 트래픽에서 SSH를 분류하는 연구는 그동안 많이 진행되어 왔지만 같은 SSH 프로토콜을 사용하면서 각기 다른 목적으로 사용되는 플로우를 분류하는 연구는 아직까지 많이 진행되어 있지 않다. 본 논문에서는 SSH 트래픽을 세 가지(정상 SSH, SFTP, SSH 터널링)의 사용패턴으로 분류하는 것이 목적이므로 SSH 트래픽을 플로우 단위로 수집 및 분류했다.

Ⅲ. SSH 프로토콜의 개요

SSH 프로토콜은 RSA 암호 메커니즘을 사용하여 암호화 호스트 인증을 통해 클라이언트와 서버 사이의 안전한 통신 채널을 제공한다. SSH의 강점은 공개키 기반의 암호화 방식을 사용하여 안전하지 않은 통신 채널을 갖고 있는 사용자들 간에 보다 안전한 암호화 통신을 해주기 때문에, 악의적인 공격자가 스니핑 도구를 이용해 사용자의 아이디와 패스워드를 쉽게 가로채지 못하게 한다는 것이다. SSH는 telnet과는 달리 강력한 인증과 X11 연결을 제공한다[5,6].

SSH 프로토콜은 다음과 같은 세 계층으로 구성된다.

연결 계층: 가장 상위 계층[7]으로 다중 보안 연결을 맺음으로써 여러 개 채널을 생성할 수 있다. 각각의 터널은 양방향으로 데이터를 전송할 수 있다. 이 계층은 SSH 프로토콜이 터미널 세션, X11 정보 포워딩, 터널 생성을 할 수 있게 한다. SSH 클라이언트는 전역 요청을 사용하여 서버 측의 포트를 요청한다. 표준 채널 타입은 터미널 쉘, SFTP, SCP 전송 기능을 포함한다.

사용자 인증 계층: 전송 계층 위에 있는 계층[8]으로 공개키 인증 방식이나 패스워드 인증 방식으

로 클라이언트 인증을 수행한다.

전송 계층: 두 개의 호스트가 인증 중의 통신 혹은 인증 후 통신을 보장한다[9]. TCP/IP 계층 위에서 동작하며 암호화, 압축 및 무결성 보장에 사용되는 스펙이나 초기 키 값 교환과 서버 인증을 관리한다.

3.1. SSH 헤더

일반적인 트래픽에서 SSH 트래픽을 골라내는 일은 크게 어렵지 않다. SSH 프로토콜은 처음 연결을 맺을 때 헤더에 SSH 프로토콜임을 명시하고 있다. 그림 1은 WireShark로 해당 패킷을 읽어왔을 때 패킷의 헤더에서 SSH 프로토콜의 사용유무와 해당 버전을 명시하고 있다는 것을 보여주고 있다.

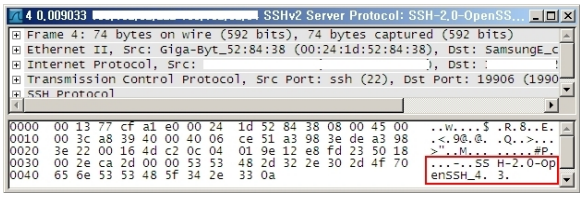


그림 1. SSH 헤더
Fig. 1. SSH Header

3.2. SSH 터널링

SSH 프로토콜은 모든 TCP(Transmission Control Protocol) 트래픽에 대한 터널링 기능을 제공한다. SSH 터널링은 SSH 서버와 SSH 클라이언트 사이의 터널을 통해 평문으로 전송될 수 있는 패킷이나 방화벽에 제한될 수 있는 모든 TCP 트래픽에 대한 기밀성과 무결성을 제공한다. 따라서 SSH 터널링은 네트워크 관리자가 볼 수 없는 트래픽을 전송하려는 목적이거나 방화벽으로 제한된 트래픽을 안전하게 전송하려는 목적으로 사용될 수 있다.

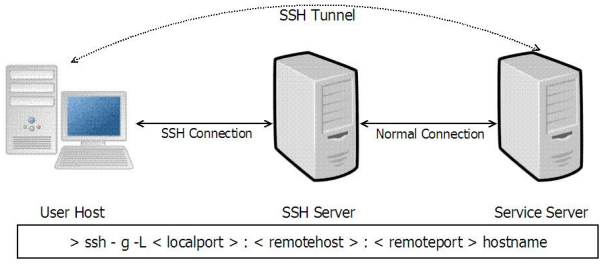


그림 2. 터널링의 개념도와 명령어
Fig. 2. Concept of SSH tunneling and command line

그림 2를 보면 사용자의 컴퓨터는 접근이 금지된 서버와 통신하려고 한다. 직접 통신할 수 없는 대신 외부에 있는 SSH 서버와 연결한 뒤 SSH 서버가 접근이 금지된 서버와의 통신을 대신해주며 데이터를 사용자의 컴퓨터에 전달해준다. 이런 식으로 금지된 서버와의 통신이 가능한 터널링 채널이 생성된다.

또 그림 2에 있는 명령어는 로컬 호스트에서 원격 호스트로의 보안 연결 설정으로 어느 목적지로든 포워딩이 가능하다. 게다가 SSH의 기본 포트 22를 사용하지 않아도 동작이 가능하므로 탐지를 피하거나 방화벽에 제한된 통신도 가능하다[10].

3.3. SFTP

SFTP(SSH File Transfer Protocol)은 SSH 프로토콜의 연결계층에서 제공하는 기능을 바탕으로 파일 접근, 파일 전송 등 파일 관리기능을 수행하는 프로토콜이다. SSH 프로토콜 2.0 버전의 파일전송 기능을 확장하여 구현되었다.

Ⅳ. SSH 트래픽의 플로우 분석

정상적인 SSH, SFTP, SSH 터널링을 사용했을 때 트래픽은 어떤 성질을 보이는 지 SSH 서버를 두고 다른 용도로 사용했을 때 각각의 트래픽을 WireShark로 수집해 페이로드, 패킷의 수, 인바운드 패킷(호스트로가 다운로드 하는 패킷), 아웃바운드 패킷(호스트가 업로드 하는 패킷)을 기준으로 분석해보았다.

4.1. 정상사용 SSH 트래픽

먼저 WireShark를 사용해 일반적인 SSH 트래픽을 수집하면 그림 3과 같이 나타난다. 그래프의 세로축은 페이로드의 크기를 나타내며 양수 값은 사용자의 호스트로 들어오는 패킷(인바운드)을 나타내고 음수 값은 SSH 서버로 전송되는 패킷(아웃바운드)를 나타낸다. 가로축은 시간을 따라 전송되는 패킷의 순서를 나타낸다.

일반적인 SSH 사용 특성상 아웃바운드 패킷은 명령어를 전송하는 목적으로 발생하기 때문에 크기가 크지 않고, 인바운드 패킷의 경우 최대전송 단위(MTU : Maximum Transmission Unit)의 패킷이 발생하지만 빈번하게 발생하지는 않는다. 그리고 인바운드 패킷의 경우 사용자 입력이 있어야만 발생되는 것을 알 수 있다.

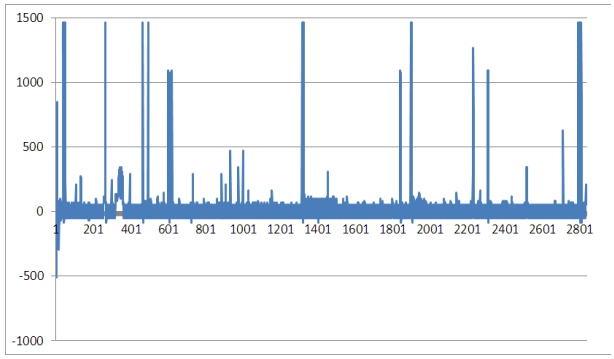


그림 3. 정상사용의 SSH 트래픽
Fig. 3. Normal SSH Traffic.

4.2. SFTP 트래픽

그림 4는 SFTP 환경에서 서버에 있는 8개의 파일을 전송받았을 때의 변화를 보여준다.

일반적인 SFTP를 사용할 때 페이로드 크기의 변화를 보면 각 파일이 전송될 때 전송중인 1460 바이트의 페이로드(MTU : Maximum Transmission Unit)가 연속적으로 발생되는 것을 볼 수 있다. 또 8개의 파일을 전송하는데 14.35초간 동작하는 동안 792개의 패킷을 발생시켰고 초당 평균 패킷 수는 55.16개이다. 파일 전송이 없을 때 아웃바운드로 일정한 크기의 페이로드를 갖는 트래픽이 발생하는 것을 볼 수 있다.

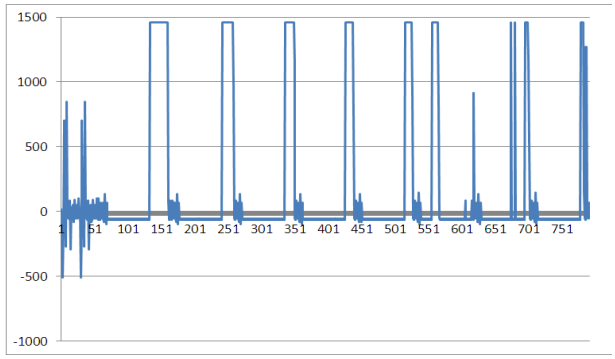


그림 4. 전형적인 SFTP 트래픽
Fig. 4. Typical SFTP traffic

4.3. 터널링 환경에서의 트래픽

8개의 파일을 전송하는데 12.5초간 동작하였고 241개의 패킷을 발생시켰다. 초당 패킷 수는 19.22 패킷으로 SFTP가 8개의 파일을 전송하는데 발생한 패킷보다 훨씬 적은 수의 패킷을 발생시켰다.

또한 파일전송 사이사이에 인/아웃바운드 패킷이 번갈아 가며 발생하는 것으로 보아 SFTP의 동작과는 다른 패턴을 보인다. 이는 터널링 환경에서는 사용자 호스트와 서버와의 연결 상태를 패킷을 주고받으며 지속적으로 확인하는 과정임을 알 수 있다.

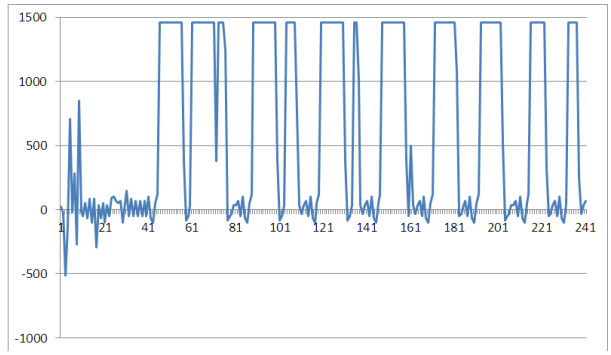


그림 5. SSH 터널링 트래픽
Fig. 5. FTP Tunneling on SSH Traffic

4.4. 트래픽 비교

위의 실험결과를 종합해 보면 다음과 같은 사실을 알 수 있다. 첫째, SFTP는 SSH 터널링을 사용한 FTP보다 많은 패킷을 발생시킨다. 이는 같은 파일을 전송시킬 때 발생되는 패킷의 수가 다르다는 것을 의미한다. 둘째, SFTP는 SSH 터널링을 통한 FTP 상태일 때와는 달리 파일 다운로드 중이 아닐 때 아웃바운드 패킷을 주로 발생시킨다. 셋째, 일반적인 SSH 트래픽은 기본적으로 사용자의 명령어 입력을 기준으로 동작하기 때문에 인/아웃바운드 값이 높지 않다. 넷째, SSH 터널링 트래픽의 경우 대량의 데이터를 전송받는 경우가 대부분이므로 정상적인 SSH 사용 일 때보다 짧은 시간에 발생되는 인바운드 트래픽이 많을 수밖에 없다.

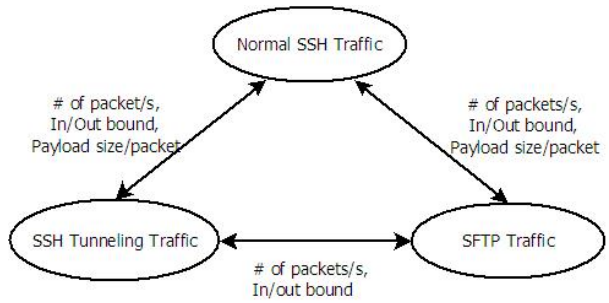


그림 6. 세가지 트래픽의 비교
Fig. 6. Comparison of 3 traffics

따라서 분석결과 정상 SSH 트래픽과 터널링, SFTP의 구별은 시간당 패킷 수, 인/아웃바운드 비율, 페이로드 크기 등으로 구별할 수 있고 터널링과 SFTP의 구별은 인/아웃바운드 비율, 시간당 패킷 수 등으로 구별할 수 있다는 것을 알 수 있다.

V. 실 험

5.1. 데이터 수집

데이터 수집은 다음과 같은 가정을 세우고 실시하였다.

가정 1: 네트워크에서 사용자는 내부에서 금지된 외부서비스를 사용하려고 한다.

가정 2: 네트워크에서 SSH 사용은 허용되어 있다.

가정 3: 금지된 외부서비스는 FTP 다운로드, HTTP 웹브라우징, BitTorrent 등 다운로드 위주의 서비스이다.

가정 4: 실험에 사용된 트래픽은 오직 서버와 클라이언트를 오가는 트래픽을 원칙으로 한다.

수집된 패킷은 플로우단위로 그룹화 되는데, 본 논문에서 플로우를 패킷 헤더의 4-tuple(Source IP, Source Port, Destination IP, Destination Port)을 기준으로 양방향으로 오가는 패킷의 집합으로 정의하였다.

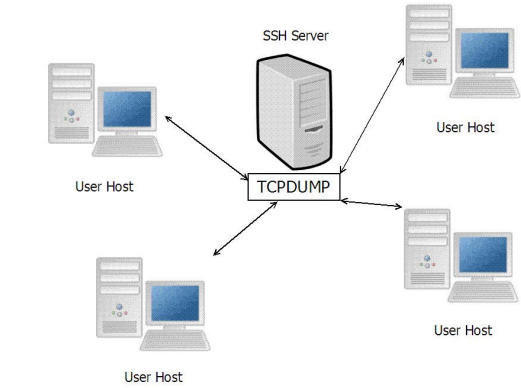
정상적인 SSH 트래픽의 경우 접속이 많은 두 대의 서버에서 TCPDUMP 프로그램을 사용해 수집한다. SFTP 트래픽의 경우 WireShark로 SFTP 데몬이 돌아가고 있는 서버와 데이터를 전송하는 트래픽을 수집한다. 터널링 트래픽의 경우 SSH 서버와 사용자 호스트 사이 터널링 명령어를 구성한 후 외부서버와 FTP 파일 전송 트래픽을 하는 과정을 WireShark로 수집하고 클러스터링을 위해 트레이닝 세트와 테스트 세트를 일정한 비율로 분리하였다.

표 1. 트래픽 수집 결과
Table 1. Traffic collection result

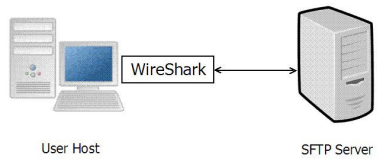
	정상사용 SSH	SFTP	SSH 터널링
Total traffic	10000 Flows	600 Flows	600 Flows
Training set	1600 Flows	100 Flows	100 Flows
Test set	8400 Flows	500 Flows	500 Flows

5.2. EM(Expectation Maximization) 클러스터링

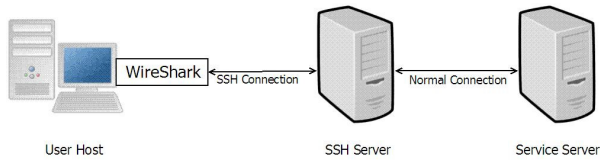
EM 알고리즘[17]은 초기 모델을 생성한 후 반복 정제과정을 통해 최적화된 모델로 만들어간다. EM 알고리즘은 반복 정제 과정을 통해 각 객체들이 혼합 모델(mixture model)에 속할 확률을 조정하여 최적의 모델을 생성해 간다.



<Normal SSH>



<SFTP>



<SSH Tunneling>

그림 7. 트래픽 수집환경
Fig. 7. Environment of traffic capture

관측할 수 있는 확률변수 X 와 관측할 수 없는 확률변수 Z , 그리고 모수가 있을 때, (X,Z)에 대한 확률 분포는 $L(\Theta;X,Z)=p(X,Z|\Theta)$으로 주어져 있다. 이때, 최대화 하려는 우도 함수는 다음과 같다.

$$L(\Theta;X)=p(X|\Theta)=\sum_{z}p(X,Z|\Theta) \quad (1)$$

EM 알고리즘은 어떠한 모수 $\Theta^{(t)}$를 입력으로 받아서 새로운 모수 $\Theta^{(t+1)}$를 찾아가는 방식인데 이러한 과정이 E(Expectation)과 M(Maximization) 단계로 나뉜다.

E(Expectation)단계에서는 $\Theta^{(t)}$가 주어졌을 때 우도의 기대값 Q를 정의한다.

$$Q(\Theta|\Theta^{(t)})=E_{Z|X,\Theta^{(t)}}[\log L(\Theta;X,Z)]$$
$$=\sum_{z}p(Z|X,\Theta^{(t)})\log L(\Theta;X,Z) \quad (2)$$

M(Maximization)단계에서는 Q를 최대화하는 새로운 모수 $\Theta^{(t+1)}$을 계산한다.

$$\Theta^{(t+1)} = \arg_{\Theta}\max Q(\Theta|\Theta^{(t)}) \quad (3)$$

실제 EM 알고리즘을 사용할 때는 모수 $\Theta^{(0)}$를 적당한 임의의 값으로 초기화한 다음, $\Theta^{(t)}$를 연속적으로 계산하면서 값이 충분히 수렴될 때 멈춘다.

Ⅵ. 실험결과

실험환경은 다음과 같다.

개발 OS : Windows XP Professional SP3
실험도구 : C++, WEKA, WireShark, TCPDUMP

실험에는 앞서 트래픽을 분석한 결과를 토대로 시간당 패킷 수, 평균 페이로드 크기, 인/아웃바운드 비율 값, 세 가지를 요소를 사용했다. 정확도를 확인하기 위해 탐지율(DR : Detection Rate), 긍정오류비율(FPR : False Positive Rate) 등을 다음과 같이 정의한다.

$$DR = \frac{TP}{TP + FN}, \quad FPR = \frac{FP}{FP + TN} \quad (4)$$

식 (4)에서 TP(True Positive)는 해당사용 트래픽을 분류알고리즘에서 정확하게 탐지한 개수를 나타내고, TN(True Negative)는 해당하지 않는 트래픽을 탐지에서 제외한 개수를 나타낸다. 또, FP(False Positive)는 해당사용 트래픽을 다른 사용의 트래픽으로 잘못 탐지한 개수를 나타낸다. FN(False Negative)는 해당사용의 트래픽이나 이를 탐지하지 못한 개수를 나타낸다.

표 1의 데이터를 플로우별 시간당 패킷 수, 평균 페이로드 크기, 인/아운바운드 패킷 비율 값, 세 가지 요소를 사용하여 EM 클러스터링을 실시하고 각각의 TP, TN, FP, FN을 각각 구한다음 탐지율(DR)과 긍정오류비율(FPR)을 정리하여 다음과 같은 결과를 얻었다.

표 2. EM 클러스터링 결과
Table 2. EM clustering result

	DR	FPR
정상사용 SSH	0.9960	0
SFTP	0.9780	0.0042
SSH 터널링	0.9900	0.0012

Ⅶ. 결론 및 향후과제

인터넷 응용 트래픽 분석은 운용관점에서 매우 중요하다. 기존의 시그니쳐 분석 방법은 널리 사용되고 있지만, 많은 한계점과 문제점을 가지고 있다. 특히 암호화된 트래픽에 대해 기존의 시그니쳐 분석 방법은 트래픽의 분류가 불가능하며 시그니쳐를 유지 및 관리하는 측면에 있어서도 그 성능을 보장하기 어렵다.

본 논문에서는 암호화된 트래픽의 한 종류인 SSH 프로토콜에 대한 분류 방법을 제안하고 실험하여 97%이상의 정확성을 보이는 분류가 가능함을 보였다.

하지만 이러한 방법이 널리 사용되기 위해서는 SSH 프로토콜 뿐만 아니라 좀 더 다양한 프로토콜에 대한 연구나 좀 더 다양한 네트워크 환경에서의 연구가 필요하다.

SFTP와 SSH 터널링 트래픽은 정상사용 SSH 트래픽과는 다른 패턴을 보인다. 특정 어플리케이션마다 트래픽 식별 하려면 각각 다른 방법을 사용해야 한다. 암호화된 트래픽 식별은 높은 탐지율을 위해 더 많은 트래픽 패턴의 연구가 필요하다.

참 고 문 헌

[1] Internet Assigned Numbers Authority (IANA), Retrieved Jun., 15., 2012., from http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml

[2] A. W. Moore and K. Papagiannaki, “Toward the accurate identification of network applications,” Passive and Active Network Measurement, Lecture Notes in Computer Science Volume 3431, 2005, pp 41-54

[3] A. Madhukar and C. Williamson, “A longitudinal study of p2p traffic classification,” *in Proc. IEEE Int. Symposium on Modeling, Analysis, and Simulation*, Sept. 2006. pp. 179 – 188.

[4] S. Sen, O. Spatscheck, and D. Wang, “Accurate, scalable in-network identification of p2p traffic using application signatures,” *in Proc. ACM Int. Conf. World Wide Web*, New York, USA, May. 2004. pp. 512 – 521.

[5] SSH FAQ, Retrieved Jun., 15., 2012., from http://www.rz.uni-karlsruhe.de/ ig25/ssh-faq/.

[6] D. J. Barett and R. E. Silverman, *SSH, The Secure Shell: The Definitive Guide*, O'Reilly, 2001.

[7] RFC4254, Retrieved Jun., 20., 2012., from http://tools.ietf.org/html/rfc4254.

[8] RFC4252 Retrieved Jun., 20., 2012., from http://tools.ietf.org/html/rfc4252.

[9] RFC4253 Retrieved Jun., 20., 2012., from http://tools.ietf.org/html/rfc4253.

[10] F. Dijkstra, A. Friedl, *Specification of advanced features for a multi-domain monitoring infrastructure*, Feb. 2010. from http://www.geant.net/Media Centre/Media Library/Pages/Deliverables.aspx.

[11] P. Haffner, S. Sen, O. Spatscheck, and D. Wang, "ACAS: Automated construction of application signatures," *in Proc. ACM SIGCOMM Workshop on Mining Network Data*, New York, USA, Aug. 2005. pp. 197–202.

[12] W. Li, M. Canini, A. W. Moore, and R. Bolla, "Efficient application identification and the temporal and spatial stability of classification schema," *Computer Networks*, vol. 53, no. 6, pp. 790–809, Apr. 2009.

[13] L. Bernaille and R. Teixeira, "Early recognition of encrypted applications," *in Proc. Int. Conf. Passive and Active Measurement*, Apr. 2007. pp. 165-175.

[14] C. Wright, F. Monrose, and G. M. Masson, "HMM profiles for network traffic classification," *in Proc. ACM Workshop on Visualization and Data Mining for Computer Security*, Oct. 2004. pp. 9–15.

[15] C. V. Wright, F. Monrose, and G. M. Masson, "On inferring application protocol behaviors in encrypted network traffic," *J. Mach. Learn. Res.*, vol. 7, pp. 2745–2769, 2006.

[16] F. Palmieri and U. Fiore, "A nonlinear, recurrence-based approach to traffic classification," *Computer Networks*, vol. 53, no. 6, pp. 761–773, Apr. 2009

[17] C. Fraley and A. E. Raftery, "How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis," *The Comput. J.*, vol. 41, no. 08, pp. 578-588, 1998.

[18] Hyunuk Kim, Ha Yoon Song, "A Study on Characterizing the Human Mobility Pattern with EM(Expectation Maximization) Clustering", Korea Computer Congress, vol.38, no. 1(B), pp. 222-225, Jun. 2011.

[19] Sung-ho Yoon, Myung-sup Kim, "A Study of Performance Improvement of Internet Application Traffic Identification using Flow Correlation", THE JOURNAL OF KOREA INFORMATION AND COMMUNICATIONS SOCIETY, vol. 36, no. 6, pp. 600-607, Jun. 2011.

[20] Sang-woo Lee, Hyun-shin Lee,Mi-jung Choi, Myung-sup Kim, "Real-time Identification of Skype Application Traffic using Behavior Analysis", THE JOURNAL OF KOREA INFORMATION AND COMMUNICATIONS SOCIETY, vol. 36, no. 2, pp. 131-140, Feb. 2011.

[21] WireShark, Retrieved Aug., 20., 2012., from http://www.wireshark.org/

[22] WinPcap, Retrieved Aug., 20., 2012., from http://www.winpcap.org/

[23] TCPDUMP, Retrieved Aug., 20., 2012., from http://www.tcpdump.org/

[24] WEKA, Retrieved Aug., 20., 2012., from http://www.cs.waikato.ac.nz/ml/weka/

김 경 륜 (Kyoung-lyoon Kim)

2009년 8월 경북대학교 컴퓨터공학과 졸업
2011년 8월 고려대학교 정보경영공학전문대학원 석사 수료
<관심분야> 네트워크 보안, 트래픽 분석

김 명 섭 (Myung-sup Kim)

1998년 2월 포항공과대학교 전자계산학과 졸업
2000년 2월 포항공과대학교 컴퓨터공학과 석사
2004년 2월 포항공과대학교 컴퓨터공학과 박사
2004년~2006년 Post-Doc., Dept. of ECE, Univ. of Torronto, Canada
2006년~현재 고려대학교 컴퓨터정보학과 부교수
<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석, 멀티미디어 네트워크

김 형 중 (Hyoung-joong Kim)

1978년 서울대학교 제어계측공학과 공학사
1986년 서울대학교 제어계측공학과 공학석사
1989년 서울대학교 제어계측공학과 공학박사
1990년~2006년 강원대학교 교수
2006년~현재 고려대학교 정보경영전문대학원 교수
<관심분야> Parallel Computing, Image Hashing, Data Compression, Steganography