NetFlow 데이터를 이용한 실시간 네트워크 트래픽 어노멀리 검출 기법

강 구 홍[†] · 장 종 수[††] · 김 기 영[†††]

요 약

최근 알려지지 않은 공격(unknown attack)으로부터 네트워크를 보호하기 위한 네트워크 트래픽 어노멀리(anomaly) 검출에 대한 관심이 고조되고 있다. 본 논문에서는 캠퍼스 네트워크의 보드라우터(border router)의 NetFlow 데이터로 제공되는 초당비트수(bits per second)와 초당플로수(flows per second)의 상관관계를 단순회귀분석을 통하여 새로운 어노멀리 검출 기법을 제시하였다. 새로이 제안된 기법을 검증하기 위해 실지 캠퍼스 네트워크에 적용하였으며 그 결과를 Holt-Winters seasonal(HWS) 알고리즘과 비교하였다. 특히, 제안된 기법은 기존 RRDtool에 통합시켜 실시간 검출이 가능하도록 설계하였다.

A Real-Time Network Traffic Anomaly Detection Scheme Using NetFlow Data

Koo-Hong Kang[†] · Jong-Soo Jang[††] · Ki-Young Kim[†††]

Abstract

Recently, it has been sharply increased the interests to detect the network traffic anomalies to help protect the computer network from unknown attacks. In this paper, we propose a new anomaly detection scheme using the simple linear regression analysis for the exported NetFlow data, such as bits per second and flows per second, from a border router at a campus network. In order to verify the proposed scheme, we apply it to a real campus network and compare the results with the Holt-Winters seasonal algorithm. In particular, we integrate it into the RRDtool for detecting the anomalies in real time.

키워드 : 침입탐지(Intrusion Detection), 어노멀리(Anomaly), 보안(Security)

1. 서 론

오늘날 산업, 정부, 그리고 심지어 일반 개인 생활이 인터넷에 점점 더 의존해 감에 따라 네트워크를 통한 정보 흐름에 관한 중요성이 한층 강조되고 있다. 그러나 해커들에 의한 네트워크 혹은 주요 서버 컴퓨터에 대한 침입이나 공격은 네트워크를 마비시킴으로서 전자 상거래 서비스 중단과 같은 심각한 경제적 손실뿐만 아니라 인터넷 서비스 중단에 따른 극심한 사회적 혼란을 초래하고 있다. 이러한 공격으로부터 네트워크를 보호하기 위한 침입탐지시스템(IDS : Intrusion Detection System) 기술은 대부분 잘 알려진 공격 시그너처(signature)를 기준으로 패턴 매칭에 의해 공격을 검출해 낸다[1-3]. 그러나 이러한 기존 방법들은 알려지지 않은 공격(일명 어노멀리(anomaly)라고 함)으로부터 네트워크를 보호하는 것은 불가능하다[4-9].

어노멀리 검출을 위한 방법은 네트워크 기반과 호스트 기반으로 이루어진다. 초기 어노멀리 검출은 호스트 기반으로서 유닉스 서버 혹은 운영체제 서비스의 오류를 이용한 공격을 검출하였다. 이들 공격은 시스템 콜의 정상적인 패턴으로부터 벗어나며, 어노멀리 검출은 이와 같은 비정상적인 시스템 콜 패턴을 확인하게 된다[5]. 한편 네트워크 기반은 네트워크 트래픽의 정상 사용 패턴(normal usage pattern)을 인지한 후 트래픽을 모니터링하여 이들 정상 사용 패턴을 벗어나는 네트워크 트래픽을 어노멀리로 검출한다[4,6]. 네트워크 트래픽 어노멀리는 네 가지 카테고리, 즉 공격, 하드웨어 장애, 사용자 폭주(flash crowds), 그리고 측정 실패 등으로 분류한다[7,8]. 이들 네트워크 트래픽 어노멀리를 빠르고 정확하게 판단하는 것은 네트워크의 효율적인 운영에 있어서 매우 중요하다.

네트워크 트래픽 어노멀리 검출 방법은 크게 두 가지 유

† 정 회 원 : 서원대학교 컴퓨터정보통신공학부 조교수
†† 정 회 원 : 한국전자통신연구원 네트워크보안그룹 그룹장
††† 정 회 원 : 한국전자통신연구원 네트워크보안그룹 팀장
논문접수 : 2004년 10월 5일, 심사완료 : 2004년 11월 4일

형으로 분류할 수 있다. 첫 번째 방법은 패킷의 헤드 정보 즉 TCP/IP 프로토콜 속성을 모니터링하여 이를 기반으로 어노멀리를 검출한다. 이러한 방법은 오늘날 대부분의 공격들이 비정상적인 TCP 플래그 혹은 IP 옵션, 유효하지 않은 일련번호(sequence number), 잘못된 체크섬(checksum), 위장된 주소(spoofed address) 등을 이용하기 때문이다. 두 번째 방법은 네트워크 트래픽을 타임 시리즈로 모델링하고 이 모델링으로부터 정상 트래픽을 정의하기 위해 통계적 편차(statistical deviations)를 결정한 다음 이들로부터 벗어나는 트래픽을 네트워크 어노멀리로 검출하는 방법이 있다. 특히, 실시간으로 네트워크 트래픽 어노멀리를 검출하기 위해 Cricket[15]/RRDtool[14] 툴에 지수 평활(exponential smoothing)과 Holt-Winters seasonal(HWS) 알고리즘을 통합시킨 연구결과[9]는 실현 가능성이 매우 우수한 것으로 보고되고 있다.

본 논문에서는 두 번째 방법인 순수 트래픽 패턴에 의존한 어노멀리 검출에 초점을 맞춘다. 이와 같은 방법은 기본적으로 네트워크 상의 트래픽 유형에 대한 풍부한 자료가 뒷받침되어야 한다. 그러나 아직 국내에서는 캠퍼스 네트워크 규모 이상에서 입·출력 트래픽의 특성이나 유형 그리고 외부로부터의 공격에 대한 정확한 자료가 보고된 바 없다. 따라서 먼저 캠퍼스 네트워크의 보드라우터(border router)의 입·출력 트래픽 특성을 조사하고 이들 라우터로부터 제공되는 NetFlow 데이터인 초당비트수(bps : bits per second)와 초당플로수(fps : flows per second)의 상관관계를 단순회귀분석을 통하여 새로운 어노멀리 검출 기법을 제시하였다. 새로이 제안된 기법을 검증하기 위해 실지 캠퍼스 네트워크에 적용하였으며 그 결과를 HWS 알고리즘과 비교 분석하였다. 특히, 제안된 기법은 기존 RRDtool에 통합시켜 실시간 검출이 가능하도록 설계하였다.

서론에 이어 제2장에서는 NetFlow 데이터를 수집하기 위한 환경과 캠퍼스 보드라우터 입·출력 트래픽 특성을 기술하고 제3장에서는 수집된 NetFlow 데이터를 사용한 새로운 네트워크 트래픽 어노멀리 검출 기법을 제안한다. 제4장에서는 실지 캠퍼스 네트워크에 적용한 결과 제시하고 HWS 알고리즘을 적용한 결과와 비교 분석한다. 제5장에서는 향후 연구 방향에 대해 언급하고, 마지막으로 제6장에서 결론을 맺는다.

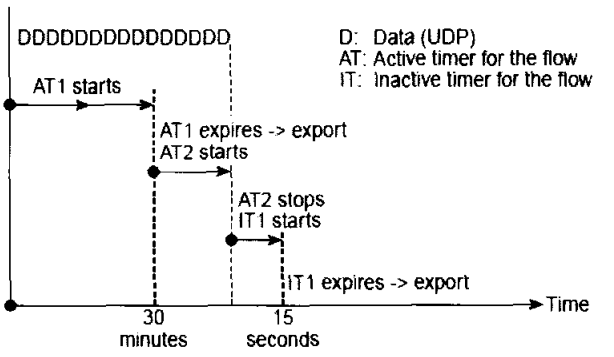
2. 캠퍼스 네트워크의 보드 라우터의 입·출력 트래픽 특성

서론에서 언급한 바와 같이 네트워크 트래픽 패턴에만 기초한 어노멀리 검출 기법은 해당 네트워크의 입·출력 트래픽 유형을 정확히 파악하는 것이 매우 중요하다. 본 장에서는 캠퍼스 네트워크의 보드라우터의 입·출력 트래픽 특징을 알아본다.

2.1 보드라우터 트래픽 정보 수집을 위한 환경

우리가 고려할 캠퍼스 네트워크는 클래스C 네트워크 9개와 /21로 수퍼넷팅(supernetting)된 하나의 네트워크로 이루어져 있으며, 외부 망과는 메트로이더넷(metro-ethernet)으로 연결되어 90 Mbps를 사용하게 설정되어 있다. 본 절에서는 외부 망과 연결된 보드라우터에 NetFlow를 활성화시켜 캠퍼스 네트워크로 입·출력되는 트래픽 형태를 조사 및 분석하였다. 시스코(Cisco)의 flow는 하나의 소스와 목적지 사이 단방향 패킷 스트림으로 정의된다. 이때 소스 및 목적지 IP 주소와 포트 주소, 그리고 계층 3 프로토콜 타입, ToS(Type of Service) 타입, 라우터의 입력 인터페이스에 따라 하나의 flow가 정의된다. NetFlow 데이터는 네트워크 관리 및 설계뿐만 아니라 과금(billing)에 이르기까지 광범위하게 사용되고 있다[13].

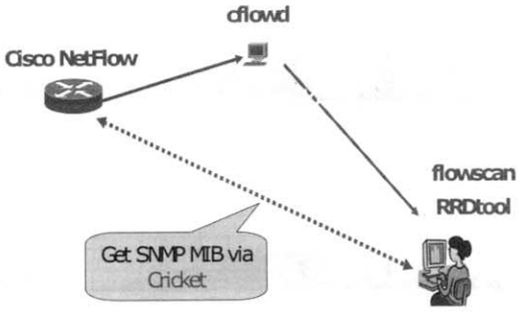
Flow는 라우터의 NetFlow 캐쉬에 저장되는데 (그림 1)에 보여지는 바와 같이 활성 및 비활성 타이머가 만료되면 UDP 데이터그램 패킷에 실려 flow 데이터가 외부로 전달된다. 한편 (그림 1)의 예와 같이 해당 flow의 데이터가 입력되지 않으면 활성 타이머가 멈추고 비활성 타이머가 시작된다. 따라서 하나의 ftp 연결을 통해 데이터 송·수신이 이루어져도, 여러 개의 flow가 존재하게 된다.



(그림 1) NetFlow의 타이머 값과 송출되는 flow 데이터의 상관관계(활성 타이머(AT : Active Timer) = 30분, 비활성 타이머(IT : Inactive Timer = 15초)

시스코 NetFlow 데이터를 수집하고 이들 수집된 데이터를 분석하는 다양한 소프트웨어 툴들이 존재한다. 본 연구에서는 (그림 2)에서 보여지는 바와 같이 비상업용 툴들을 사용하여 이들 NetFlow 데이터를 분석한다. 먼저, 라우터로부터 전송된 NetFlow 데이터를 flow-tools 툴을 사용하여 캡쳐하고 저장된 파일을 cflowd 형식으로

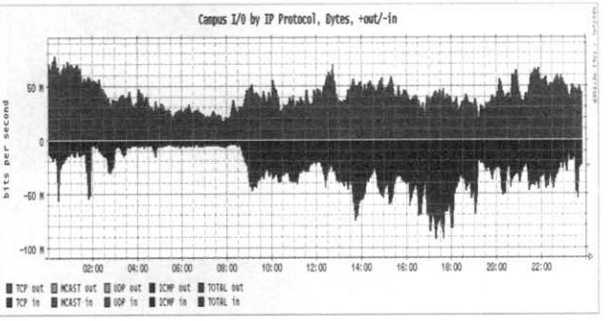
변환한다[12]. 두 번째로 flowscan 툴을 사용하여 flow-tools에 의해 저장된 파일을 분석하고 RRDtool을 호출하여 데이터베이스화 한다[14]. 마지막으로 RRDtool을 사용하여 분석에 필요한 다양한 그래프 및 파일을 데이터베이스화한다. 한편, 보드라우터의 SNMP(Simple Network Management Protocol), MIB(Management Information Base) 정보를 수집하기 위해 cricket 툴을 사용한다. 본 연구에서는 보드라우터 NetFlow 정보를 송출하기 위해 활성 타이머는 1분, 그리고 비활성 타이머는 20초로 각각 설정하고 5분 간격으로 RRDtool을 사용하여 수집된 정보를 GUI(Graphic User Interface)로 나타낸다.



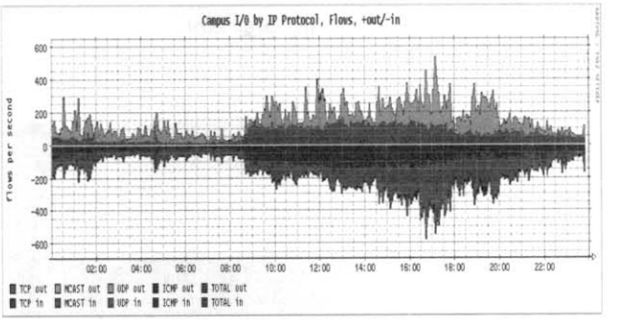
(그림 2) NetFlow 정보 및 SNMP MIB 수집

2.2 보드라우터의 입·출력 트래픽

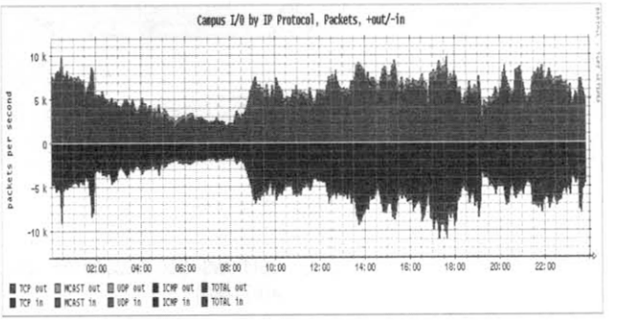
(그림 3)은 캠퍼스 보드라우터에서 평일 하루 동안 수집된 트래픽 형태((a) bps, (b) fps, (c) 초당패킷수 (pps : packets per second))를 보여준다. (그림 3)의 y축 0을 중심으로 양의 방향으로는 출력 트래픽, 그리고 음의 방향으로는 입력 트래픽을 각각 보여준다. (그림 3)을 통해 fps와 pps 입·출력은 상·하 대칭적 패턴을 보여주나 bps는 비대칭적이다. 이러한 현상은 TCP의 ACK 패킷등으로 인해 pps는 어느 정도 입·출력 대칭성을 보여주게 된다. 한편 (그림 3)으로부터 하루 동안 트래픽 유형을 파악할 수 있다. 즉 오전 9시부터 트래픽의 흐름이 급격하게 증가하고 오후 6시부터는 서서히 트래픽이 감소함을 확인할 수 있다. (그림 3) (b))를 통해 트랜스포트(transport) 계층 프로토콜인 TCP 및 UDP flow와 미세한 ICMP flow를 확인할 수 있다. 그러나 트래픽 양으로 볼 때는((그림 3) (a) 참조) TCP 트래픽이 전체 트래픽의 대부분을 차지함을 확인할 수 있다. 이것은 UDP 및 ICMP의 경우 TCP에 비해 데이터의 양이 절대적으로 작다는 것을 의미한다. 따라서 UDP 및 ICMP 트래픽 양이 증가할 경우, 이것은 네트워크 어노멀리 임을 강하게 확신할 수 있을 것이다. 궁극적으로 외부 공격으로부터 네트워크를 보호하기 위해 입력 TCP 트래픽의 bps와 fps 트래픽 분석에 초점을 맞춘다.



(a) 초당 비트 수 bps(bits per second)



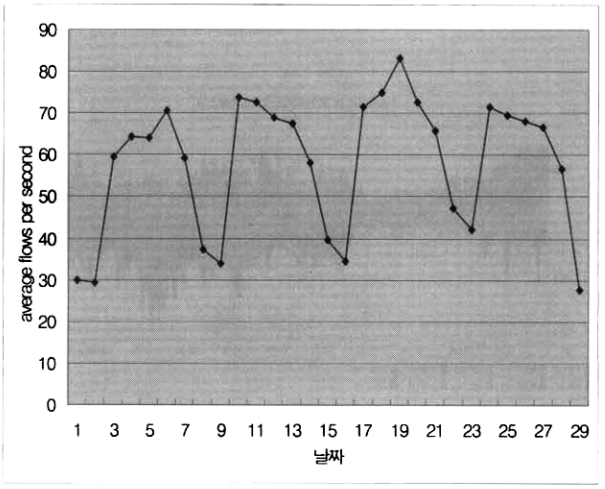
(b) 초당 flow 수 fps(flows per second)



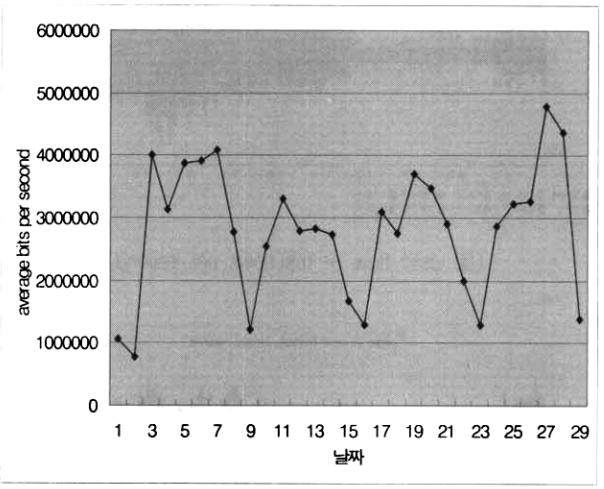
(c) 초당 패킷 수 pps(packets per second)

(그림 3) 평일 하루 동안 수집된 트래픽 형태

(그림 4)는 한 달 (2004년 8월 28일 토요일~2004년 9월 25일 토요일) 동안 수집된 하루 동안의 입력 TCP 트래픽의 평균 bps와 fps의 변화를 보여준다. 그림을 통해 1주일을 주기로 비교적 규칙적인 패턴이 반복됨을 확인할 수 있으며, 월요일에서 금요일 사이의 트래픽 양과 주말 트래픽 양은 상당한 차이를 보여준다. 즉 토요일과 일요일의 경우 하루 평균 초당 30~45 flows 그리고 초당 1~2메가 비트(mega bits), 월요일~금요일의 경우 하루 평균 초당 55~80flows 그리고 초당 3~4메가 비트로 평균 두 배 정도의 트래픽 양의 차이를 보여준다.



(a) 평균 fps 변화 추이



(b) 평균 bps 변화 추이

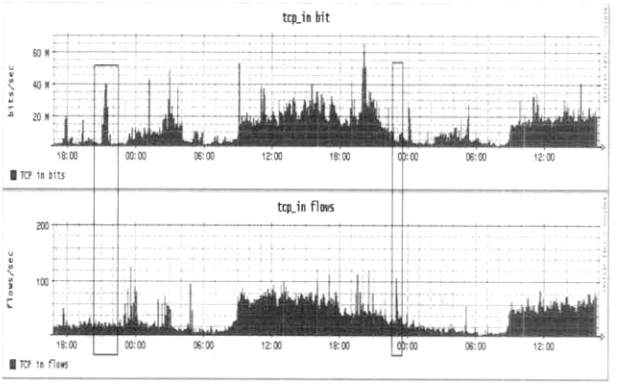
(그림 4) 한 달 동안 수집된 fps(flows per second)와 bps (bits per second)의 변화 추이

3. NetFlow 데이터를 이용한 새로운 어노멀리 검출 기법

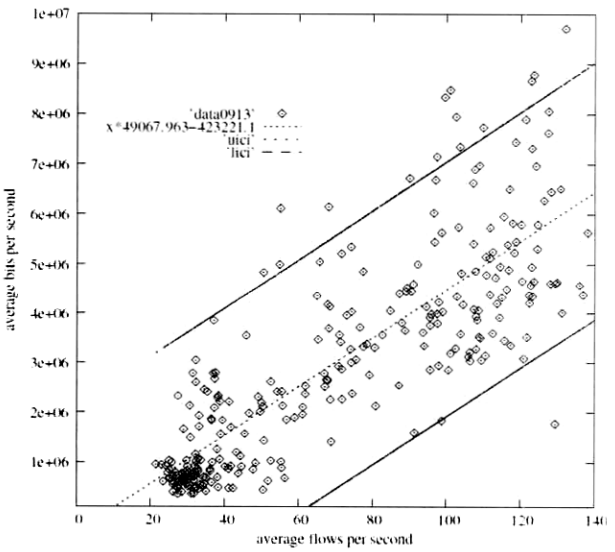
3.1 동기

(그림 5)는 5분마다 NetFlow로부터 수집된 입력 TCP의 bps와 fps 트래픽 형태를 보여준다. 이들 두 트래픽 파라미터에 대한 하루 동안 측정된 값의 산점도(scatter diagram)는 (그림 6)과 같다. (그림 5)와 (그림 6)으로부터 bps와 fps는 상호 의존적이며, (그림 5)로부터 네트워크 트래픽 어노멀리와 관련하여 다음 두 가지 사실을 확인할 수 있다. 먼저, (i) fps에 비해 bps가 지나치게 큰 경우이며, 두 번째는 (ii) bps에 비해 fps가 지나치게 큰 경우이다. 서비스거부공격(DoS : denial-of-service)은 과도한 트래픽을 희생자(victim)에 전송하여 CPU, 메모리, 혹은 네트워크 자원을 고갈시켜 정상적인 사용자들이 접근하지 못하게 하는 플러딩(flooding) 공격이 대표적이다 [15]. 이와 같은 네트워크 과부하는 짧은 시간에 가능한 최대한 많은 트래픽을 전송함으로서 얻어지며 상기 (i)에 해당된다.

한편, 오늘날 잘 알려진 TCP SYN 공격은 희생자의 열려진 TCP 포트로 계속적인 연결을 시도함으로서 희생자의 메모리를 고갈시키는 방법으로 상기 (ii)에 해당된다. 즉 fps는 많은 반면, 실질적인 트래픽 양 즉 bps는 을 증가하지 않을 것이다.



(그림 5) 2일간 측정된 입력 TCP의 bps와 fps의 트래픽 관계



(점선 : 단순 선형회귀선, 실선 : 개별사례 95% 신뢰구간 상한선과 하한선)

(그림 6) 2004년 9월 13일 하루 동안 수집된 fps와 bps의 상관관계 산점도

3.2 단순 선형 회귀분석에 의한 네트워크 트래픽 어노멀리 검출 기법

3.1절에서 설명한 바와 같이 fps와 bps 사이에는 선형관계

를 가진다고 볼 수 있다. 즉 flow 수가 증가하면 트래픽의 양도 따라서 증가한다고 쉽게 예측된다. 이제 fps를 독립변수 x로 그리고 bps를 종속변수 y로 선정하여 단순선형회귀 모형으로 이들 선형관계를 다음 식 (1)과 같이 나타낼 수 있다[11].

$$y = \beta_0 + \beta_1 x \tag{1}$$

여기서, 회귀계수(regression coefficient) β_0는 상수 혹은 절편(intercept)이고 β_1은 기울기(slope)이다. 그러나 y의 모든 관찰값들이 식 (1)의 회귀식에 의해 설명되는 것은 아니다. 즉 x로부터 y 값을 예측하는데 따르는 오차가 존재하며 따라서 i 번째 관측 x_i의 예측회귀식은 다음 식 (2)와 같다.

$$\hat{y}_i = \beta_0 + \beta_1 x_i \tag{2}$$

회귀모형의 주 사용처는 특정 x에 대한 새로운 관찰값 y를 예측하는 것이다. 이제 새로운 x_0가 주어졌을 때 $100(1-\alpha)\%$ 예측 구간(prediction interval)은 다음과 같다.

$$\hat{y}_0 - t_{(\alpha/2,\,n-2)}\sqrt{MS_E\left(1+\frac{1}{n}+\frac{(x_0-\bar{x})^2}{S_{xx}}\right)} \le y_0 \le \hat{y}_0 + t_{(\alpha/2,\,n-2)}\sqrt{MS_E\left(1+\frac{1}{n}+\frac{(x_0-\bar{x})^2}{S_{xx}}\right)} \tag{3}$$

여기서, $t_{(\alpha/2,\,n-2)}$는 $n-2$ 자유도(degree of freedom) t 분포, MS_E는 잔차평균자승(residual mean square), $S_{xx} = \sum_{i=1}^{n}(x_i - \bar{x})^2$, 그리고 $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$이다.

본 연구에서는 SPSS(Statistical Package for the Social Science) 프로그램을 이용하여 식 (2)의 회귀계수와 식 (3)의 잔차평균자승을 구한다[16]. 한편, 상기 (그림 6)은 95% 신뢰구간에 대한 식 (3)에 근거한 상한선과 하한선을 각각 표시하였다. 본 논문에서는 이들 예측 구간을 벗어나는 점들에 대해 네트워크 트래픽 어노멀리를 규정하였으며 신뢰구간의 퍼센트는 네트워크 및 트래픽의 형태에 따라 튜닝할 필요가 있다. 다음 <표 1>은 2004년 9월 13일 월요일 입력 트래픽에 대해 SPSS를 이용해 구한 회귀계수이며 따라서 다음과 같은 단순회귀선을 구할 수 있다.

$$\hat{y}_i = -423221.1 + 49067.963 x_i \tag{4}$$

3.3 회귀분석 모형 검증

(그림 6)으로부터 fps와 bps는 선형적인 관계가 있음을 직관적으로 확인할 수 있다. 그러나 우리는 t 시험 혹은 F 시험 등을 통해 이들 선형관계를 보다 정확하게 검증할 수 있다. 회귀모형을 이용해 선형성을 검증하기 위해 다음 두 가설을 세운다.

$$H_0 : \beta_1 = 0$$
$$H_a : \beta_1 \neq 0$$

F 시험의 경우, 유의수준(level of significance) α에 대해 다음과 같이 결론을 내릴 수 있다.

$$\text{If } F^* \le F(1-\alpha;1,n-2), \text{ conclude } H_0$$
$$\text{If } F^* > F(1-\alpha;1,n-2), \text{ conclude } H_a$$

여기서, 유의수준 α 95%에 $n = 288$을 적용하여 F 분포표를 통해 $F(.95;1,288) = 3.84$를 구할 수 있다. <표 1>의 잔차분포표로부터 $F^* = 539.208 > 3.84$ 임으로 H_0를 기각한다. 따라서 fps와 bps 사이 선형관계가 있음을 검증한다.

회귀분석 과정에서 회귀식이 추정되면 그 결과를 받아들이기에 앞서 잔차(residual)에 대한 다음 몇 가지 가정 - (i) 잔차의 평균은 0의 값을 지닌다. (ii) 잔차는 정규분포의 형태를 취한다. 그리고 (iii) 잔차들은 서로 독립적이어야 한다. 즉 자기상관(autocorrelation)을 보이지 않아야 한다 - 이 성립하는지 확인하는 것이 필요하다.

〈표 1〉 2004년 9월 13일 월요일 입력 트래픽에 대해 SPSS를 이용해 구한 회귀계수

Coefficients[a]

Model	Unstandardized Coefficients		Standardized Coefficients	t	Sig.	95% Confidence Interval for B	
	B	Std. Error	Beta			Lower Bound	Upper Bound
1(Constant)	-423221.1	69556.716		-2.496	.013	-756958.465	-89483.783
FPS	49067.963	2113.099	.808	23.221	.000	44908.764	53227.161

a. Dependent Variable : BPS

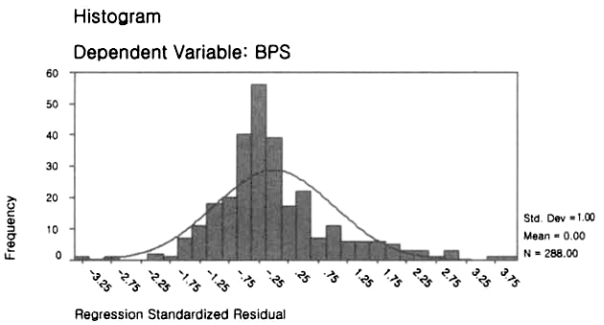
〈표 2〉 SPSS를 통해 구한 잔차분석표(ANOVA : analysis of variance)

ANOVA[b]

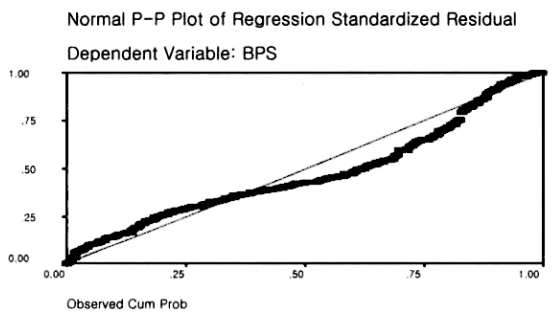
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	9.171E+14	1	9.17098E+14	539.208	.000[a]
	Residual	4.864E+14	286	1.70082E+12		
	Total	1.404E+15	287			

a. Predictors : (Constant), FPS
b. Dependent Variable : BPS

(그림 7) (a)는 잔차에 대한 히스토그램을 보여준다. 그림에서와 같이 평균이 0이고 정규분포의 형태를 취하고 있다. 보다 공식적인 잔차의 정규성 판단은 정규확률도((그림 7)(b) 참조)로부터 확인할 수 있다. (그림 9)의 x 축은 잔차의 관측된 누적확률(cumulative probability)을 그리고 y 축은 누적확률의 기대값을 나타낸다. 그림에서 45도 기울기의 대각선을 따라 균등하게 퍼져 있어 분포가 정상적이라는 것을 확인할 수 있다.



(a) 히스토그램



(b) P-P 그림

(그림 7) SPSS를 통해 구한 잔차의 히스토그램 및 P-P 그림

잔차의 동분산성은 (그림 8)과 같이 잔차산점도로부터 쉽게 판별해낼 수 있다. 즉 그림에서와 같이 fps의 증가에 따라 잔차들의 분포가 그다지 넓게 퍼져가지 않음을 확인할 수 있다. 또한 잔차의 자기상관성을 확인하기 위해 다음과 같이 Durbin-Watson 검증의 과정을 따라 정적 또는 부적 자기상관이 없다는 영가설(H_0)인 경우,

$$d_U < d < 4 - d_U : H_0 \text{ 채택}$$

을 확인한다. 여기서, 유의수준에서의 상한값 d_U는 Durbin-Watson의 d 통계량 표에서 찾는다. 본 자료에서는 5% 유의수준에 독립변수의 수 $k = 1$ 그리고 표본수 $n = 288$에 해당하는 $d_u = 1.69$를 구할 수 있다. 한편, <표 3> model summary 표로부터

$$1.69 < 1.99 < 2.31$$

을 확인할 수 있다. 따라서 우리는 자기상관이 없다는 영가설을 채택할 수 있다.

3.4 RRDtool 그래프 생성

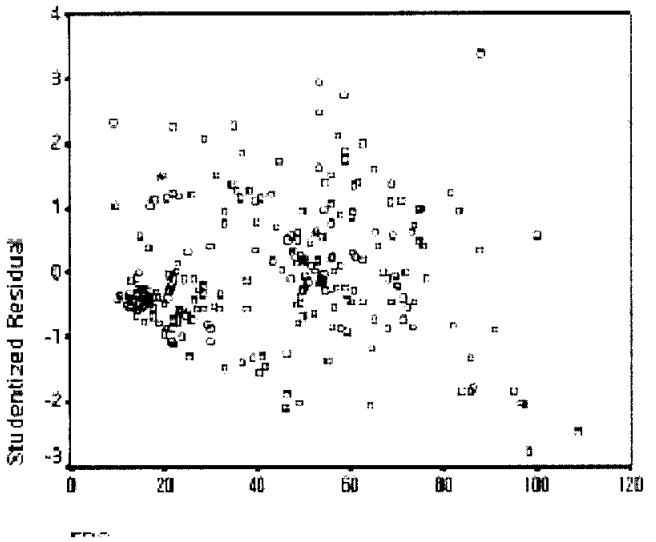
제2장에서 설명한 바와 같이 보드라우터에서 NetFlow 정보를 활성화시키고 활성 타이머를 1분 그리고 비활성 타이머는 20초로 각각 설정하였다. 이와 같이 타이머들의 값을 미세하게 조정한 이유는 가능한 트래픽 양과 flow 수의 상관관계를 밀접하게 유지하기 위함이다. 한편, 실시간 어노멀리 검출을 위해 다음 (그림 9)와 같이 식 (3)을 RRDtool 그래프 생성에 적용하였다.

〈표 3〉 SPSS에 의한 모델 요약

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Durbin-W atson
1	.808[a]	.653	.651	7.77578E+05	1.990

a. Predictors : (Constant), FPS
b. Dependent Variable : BPS



(그림 8) fps에 대한 스튜덴트 잔차(studentized residual)의 분포

```
DEF:x=/flows/rrds/tcp_monitor_dump.rrd:flows
      :AVERAGE \
DEF:y=/flows/rrds/tcp_monitor_dump.rrd:bytes
      :AVERAGE \
CDEF:Y0=49067.963,x,*,42322.1,- \
CDEF:SSE=x,71.526134,-,POWER \
CDEF:SSE1=SSE,380907.469854,/ \
CDEF:SSE2=1,288,/,1,+,SSE1,+,SQRT,1304154.89
      8775,* \
CDEF:SSE3=1.645,SSE2,* \
CDEF:upper=Y0,SSE3,+ \
CDEF:lower=Y0,SSE3,- \
CDEF:failupper=y,upper,GT,1,0,IF \
CDEF:faillower=y,lower,LT,1,0,IF \
TICK:failupper#00dd00:1:"Failures uppers" \
TICK:faillower#ff44ff:1:"Failures lowers" \
LINE2:x#000fff:"Average in flows/sec"
```

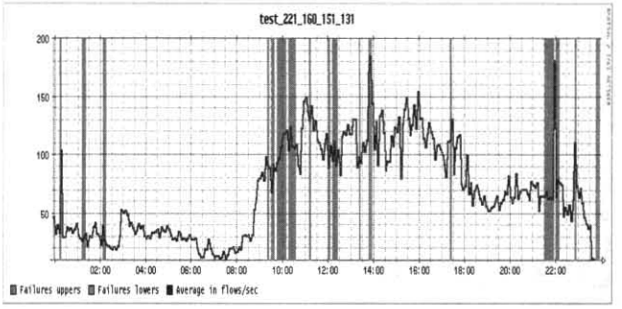
(그림 9) RRDtool 그래프 생성 명령

4. 결과 분석

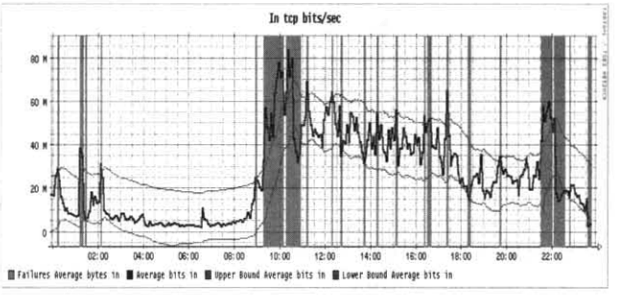
본 장에서는 캠퍼스 네트워크의 보드라우터에서 실시간으로 입력되는 트래픽을 기준으로 새로이 제안된 기법의 기능을 확인하였다. 뿐만 아니라 기존의 HWS 알고리즘을 이용한 어노멀리 검출 결과와 비교 분석하였다.

이들 검출 알고리즘들이 운영중인 실지 네트워크에 적용되기 위해서는 관련 파라미터들의 튜닝작업들이 필요하게 된다. 예를 들어, 제안된 기법의 경우 신뢰구간 설정을 위한 α 값 선정에 있으며 HWS 알고리즘을 이용한 어노멀리 검출 기법[9]의 경우 7개 파라미터(alpha, beta, gamma, window -length, failure-threshold, deltapos, deltaneg) 값을 선정해야 한다. 여기에서, window-length와 failure threshold 파라미터는 false positive 혹은 false negative 성능지수를 개선하기 위한 슬라이딩 윈도우 관련 파라미터이다. 즉 윈도우 사이즈(window-length : win)를 설정하고 이 윈도우 내에 발생된 어노멀리 개수가 문턱치 값(failure-threshold : thr) 보다 클 경우 비로소 최종 어노멀리로 정의한다. 그러나 이들 파라미터들은 네트워크 환경에 따른 많은 시행착오를 통한 튜닝작업에 의해 결정되는 것이며, 아직 이들 파라미터 설정에 관한 정확한 방법이 제시되지 못하고 있는 실정이다. 따라서 본 연구에서는 좀 더 객관적인 비교를 위해 먼저 윈도우 개념을 배제하고($win = 1, thr = 1$) 두 기법을 비교한다. 한편, HWS 알고리즘을 이용한 어노멀리 검출을 위한 나머지 5개 파라미터는 참고문헌 [9]에서 제시한 디폴트 파라미터를 사용하여 어노멀리 검출결과를 제시한다((그림 10) 참조). 궁극적으로 새로이 제안된 방법은 하나의 파라미터 α 값만 선정하면 됨으로 HWS 알고리즘에 비해 튜닝작업이 상당히 간단해 질 것이다.

(그림 10)은 2004년 9월 30일 하루 동안 캠퍼스 네트워크 보드라우터 입력 트래픽에서 측정된 어노멀리 검출 결과이며, 수직선들은 해당 시점에 어노멀리가 검출되었음을 나타낸다. (그림 10) (a)와 (b)로부터 제안된 기법을 사용한 결과와 HWS 알고리즘을 적용한 결과를 비교하면 다소 차이가 있으나 이러한 차이는 파라미터 튜닝 작업을 통해 줄여 나갈 수가 있다.



(a) 본 논문에서 제안한 기법 적용 결과 (실선은 TCP 입력 트래픽 fps)

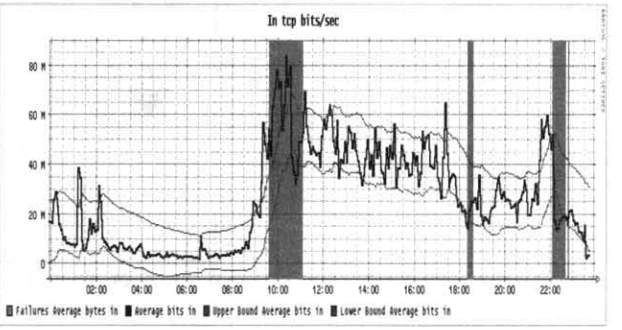


(b) HWS 알고리즘 적용 결과(윈도우 사이즈 = 1, 실선은 TCP 입력 트래픽 bps)

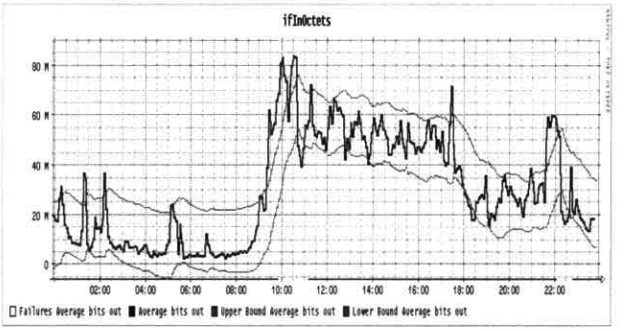
(그림 10) 2004년 9월 30일 어노멀리 검출 결과(상하 수직선 : 어노멀리 검출)

앞에서 설명한 바와 같이 false positive를 줄이기 위해서는 윈도우의 개념을 도입해야 하며 (그림 11)은 참고문헌 [9]에서 제시한 $win = 9, thr = 7$을 적용하여 HWS 알고리즘에 의한 어노멀리 검출 결과이다. 즉 윈도우사이즈 내에 7개이상의 어노멀리가 검출되면 비로소 경고를 발생하게 된다. (그림 11)(a)는 NetFlow 정보를 이용한 것이며 (그림 11)(b)는 Cricket 툴을 이용한 SNMP MIB 정보를 이용한 것이다. 이들 결과가 다소 다른 것은 NetFlow에 의한 TCP 입력 트래픽의 특성과 Cricket 툴에 의한 전체 입력 트래픽의 특성(SNMP MIB ifInOctets : 인터페이스를 통해 입력되는 전체 옥텟 수)의 미세한 차이 때문이다. 그러나 이러한 차이점 역시 관련 파라미터의 튜닝 작업으로 충분히 극복할 수 있다. 따라서 참고문헌 [9]에서 사용한 MIB 정보 대신 NetFlow 정보를 기준으로 HWS 알고리즘을 적용하여도 동일한 어노멀리 검출 결과를 얻을 수 있다. 한편 새로이 제시한 기법 역시 이러한 윈도우 개념을 쉽게 도입할 수 있으며 그 결과 또한 (그림 11)과 동일하게 나타날 것이라고 (그림 10)(a)로부터 쉽게 예측된다.

한편, (그림 4)를 통해 확인한 바와 같이 트래픽은 한 주를 주기로 변화되며, 주말 트래픽은 특히 다른 요일에 비해 차이가 심하다. 따라서 주말은 주중과는 다른 관련 파라미터를 선정해야 한다. 주중의 경우, 요일별로 다소 차이는 있으나 이들 차이점들이 일정한 규칙을 갖는 것은 아니다. 즉 매 주 월요일이라고 해서 다른 요일에 비해 트래픽 양이 항상 많은 것은 아니다. 따라서 주중에는 동일한 파라미터를 적용하였다.



(a) NetFlow에 의한 TCP 입력 트래픽 정보를 이용한 결과 (윈도우 사이즈 = 9, 문턱치 = 7, 실선은 TCP 입력 트래픽 bps)



(b) Cricket 툴에 의한 SNMP MIB ifInOctetes를 이용한 결과(윈도우 사이즈 = 9, 문턱치 = 7, 실선은 SNMP MIB ifInOctets [17])

(그림 11) 2004년 9월 30일 HWS 알고리즘을 이용한 어노멀리 검출 결과(상하 수직선 : 어노멀리 검출)

5. 향후 연구 방향

본 논문을 통해 제안된 방법은 요일별로 고정된 단순회귀모형의 파라미터를 사용하였다. 그러나 캠퍼스 네트워크의 특성 상 네트워크 트래픽은 계절별로 상당한 차이를 나타낼 것이며 학사 행정상의 이유로 flash crowd가 존재할 것이다. 특히 계절별 차이는 HWS 알고리즘과 같이 타임시리즈 개념 도입이 절실하다[10]. 즉 지수 평활화(exponential smoothing)를 이용해 t 요일의 단순회귀모형 파라미터 $\beta_{0,t}$는 다음과 같이 사용할 수 있다.

$$\beta_{0,t} = \gamma\beta_{0,t-w} + (1-\gamma)\beta_{0,t-m}$$

여기서, m은 계절 사이클의 주기이며 w는 한 주 사이클의 주기이며 γ는 0과 1 사이 값을 갖는다. 따라서 $\beta_{0,t}$는 지난 주 사이클의 동일 시점과 지난 계절 사이클의 동일 시점을 각각 참조하게 된다. 한편 단순회귀모형 $\beta_{1,t}$ 역시 동일 방법을 적용할 수 있다. 그러나 이러한 타임시리즈의 개념 도입은 외국의 사례와 같이 2, 3년 이상의 꾸준한 네트워크 트래픽의 데이터 베이스화와 함께 오프라인 상에서 정확한 네트워크 어노멀리 검출 툴들이 존재해야만 할 것이다.

또한 제안된 기법의 타당성을 검증하기 위해 HWS 알고리즘과의 단순 비교 분석에 머물렀으나 보다 정확한 검증을 위해 네트워크 공격 툴들을 이용하거나 어노멀리 시나리오를 포함한 네트워크 트래픽 데이터베이스를 확보하여 적용한 결과를 제시할 예정이다.

6. 결 론

본 논문에서는 캠퍼스 네트워크의 보드라우터의 NetFlow 데이터로 제공되는 bps와 fps의 상관관계를 단순회귀분석을 통하여 새로운 어노멀리 검출 기법을 제시하였다. 새로이 제안된 기법을 검증하기 위해 실지 캠퍼스 네트워크에 적용하였으며 그 결과를 HWS 알고리즘과 비교하였다. 특히, 제안된 기법은 기존 RRDtool에 통합시켜 실시간 검출이 가능하도록 설계하였다. 새로이 제안된 기법은 기존의 HWS 알고리즘과 비교해 성능 면에서는 유사한 반면 설정해야 할 파라미터 개수가 적어 네트워크에 적용하기 위한 튜닝 작업이 용이하다. 즉 기존의 HWS 알고리즘의 경우 지수평활화 관련 파라미터 세 개와 신뢰구간 밴드 설정을 위한 하나의 파라미터를 결정해야 하는 반면 제안된 기법은 신뢰구간 퍼센트만 결정하면 된다. 이러한 튜닝 작업의 용이성은 망 관리자에게 상당한 편리성을 제공하게 될 것이다.

본 논문은 NetFlow 데이터를 네트워크 트래픽 어노멀리 검출에 적용한 최초의 연구 결과다. 그러나 제5장에서 언급한 바와 같이 아직 시작 단계에 불과하며 앞으로 이루어질 많은 연구들의 출발점이 될 것을 희망한다.

참 고 문 헌

[1] M. Roesch, "Snort Lightweight Intrusion Detection for Networks," Proc. USENIX LISA'99 pp.101-109, 1999.

[2] H. Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of intrusion-detection systems," Computer Networks, Vol. 31, No.8, pp.805-822, 1990.

[3] F. Gong, "Next Generation Intrusion Detection System (IDS)," IntruVert Networks Report, 2002.

[4] Mattew V. Mahoney, and Philip K. Chan, "Learning Nonstationay Models of Normal Network Traffic for Detecting Novel Attacks," *in Proceedings of SIGKDD'02*, 2002.

[5] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, "A Sense of Self for Unix Processes," *in Proceedings of 1996 IEEE Symposium on Computer Security and Privacy*, 1996.

[6] Eleazar Eskin, "Anomaly Detection over Noisy Data using Learned Probability Distributions," *in Proceedings of ICML-2000*, 2000.

[7] Paul Barford and David Plonka, "Characteristics of Network Traffic Flow Anomalies," *in Proceedings of the ACM Internet Measurement Workshop*, Nov., 2001.

[8] Paul Barford, Jeffery Kline, David Plonka, and Amos Ron, "A Signal Analysis of Network Traffic Anomalies," *in Proceedings of the ACM Internet Measurement Workshop*, Nov. 2002.

[9] Jake D. Brutlag, "Aberrant Behavior Detection in Time Series for Network Monitoring," in *Proceedings of the USENIX Fourteenth system Administration Conference LISA XIV*, 2000.

[10] Peter J. Brockwell, and Richard A. Davis, *Introductionto Time Series and Forecasting*, Springer-Verlag, 1996.

[11] D. C. Montgomery, and E. A. Peck, Introduction to Linear Regression Analysis, 2nd Ed., John Wiley & Sons, Inc., 1992.

[12] D. Plonka, "Flowscan : A network traffic flow reporting and visualization tool," in *Proceedings of the USENIX Fourteenth system Administration Conference LISA XIV*, 2000.

[13] Cisco, NetFlow Services Solutions Guide, Cisco White Paper, 2001.

[14] T. Oetiker, The RRDtool manuals, http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/manual/index.html

[15] J. R. Allen, The Cricket reference guide, http://cricket.sourceforge.net/support/doc/reference.html

[16] SPSS manual, http://www.spss.com

[17] K. McCloghrie, and M. Rose, "Management information base for network management of tcp/ip based internets : Mib 2," RFC1213, 1991.

강 구 홍

e-mail : khkang@seowon.ac.kr

1985년 경북대학교 전자공학과(공학사)

1990년 충남대학교 대학원 전자공학과 (공학석사)

1998년 포항공과대학교 대학원 전자계산학과(공학박사)

1985년~1993년 한국전자통신연구소 선임연구원

1998년~1999년 한국전자통신연구원 선임연구원

2002년~2003년 한국전자통신연구원 초빙연구원

2000년~2001년 서원대학교 컴퓨터정보통신공학부 전임강사

2002년~현재 서원대학교 컴퓨터정보통신공학부 조교수

관심분야 : 성능평가, 컴퓨터 네트워크, 네트워크 보안 등

장 종 수

e-mail : jsjang@etri.re.kr

1984년 경북대학교 전자공학과(공학사)

1986년 경북대학교 대학원 전자공학과 (공학석사)

2000년 충북대학교 대학원 컴퓨터공학과 (공학박사)

1989년 7월~현재 한국전자통신연구원 네트워크보안그룹장

관심분야 : 네트워크 보안, 센서네트워크, 정책기반보안관리, QoS 등

김 기 영

e-mail : kykim@etri.re.kr

1988년 전남대학교 전산통계학과(공학사)

1993년 전남대학교 대학원 전산통계학과 (공학석사)

2002년 충북대학교 대학원 전자계산학과 (공학박사)

1988년~현재 한국전자통신연구원 보안게이트웨이연구팀 팀장

관심분야 : 네트워크 보안, 고성능 네트워크 침입탐지 및 대응기술 등