트래픽 분석을 이용한 호스트 정보 수집

이현신*, 오영석**, 이상우**, 김명섭**
*고려대학교 수학과
**고려대학교 컴퓨터정보학과
e-mail : {oshyuns, 840105, ilovejam, tmskim}@korea.ac.kr*,

Host information gathering using the traffic analysis

Hyun-Shin Lee*, Sang-Woo Lee**, Myung-Sup Kim**
*Dept. of Mathematics, Korea University
**Dept. of Computer And Information Science, Korea University

요 약

본 논문은 단말 호스트에서 발생한 트래픽 정보를 분석하여 단말 호스트의 다양한 정보를 수집하는 방법론에 대하여 기술한다. 본 논문에서는 첫째로 TCP 의 3-way handshake 중 SYN 패킷의 정보를 이용한 호스트의 운영체제를 예측하는 방법론과 해당 호스트에서 발생한 TCP 연결의 응답시간 분포를 분석하여 호스트의 네트워크 접근 밥법이 유·무선인지 분류하는 새로운 방법론을 제안한다. 분석이 완료된 호스트는 데이타베이스에 해당 호스트의 정보를 기록한다. 이는 웹을 통해 손쉽게 확인 가능하도록 하기 위함이다. 또한 하나의 호스트에서 유·무선 트래픽이 동시에 발생되었을 경우, 이에 대한 정보를 기반으로 유·무선 공유기 설치 유무를 판별할수 있도록 설계하였다.

1. 서론

현재 대기업에서부터 SO-HO(Small Office - Home Office)에 이르기까지 다양한 분야에서 네트워크 기반 시설은 널리 사용되고 있다. 그에 대한 관리의 필요성도 점차 늘어나고 있고 네트워크 관리를 위해서는 다양한 트래픽 분류가 선행되어야 한다. 본 논문에서는 단말 호스트에서 발생한 트래픽 정보를 분석하여 단말 호스트의 다양한 정보를 수집하는 방법론을 제안한다.

첫째로 TCP 의 3-way handshake 중 SYN 패킷의 정보를 이용한 호스트의 운영체제를 예측하는 방법론과 둘째로 단말 호스트에서 발생한 TCP 연결의 응답시간 분포를 분석하여 호스트의 네트워크 접근 방법이 유·무선인지 분류하는 방법론을 제안한다.

선행 연구에서는 분류 대상을 호스트로 선정하였고 ICMP 를 이용한 응답시간에 따른 분류를 하였다. 이는 공유기가 설치된 호스트에 대해서는 판별이 불가능하였으며, ICMP 의 echo reply 를 공유기에서 처리하기 때문에 외부 네트워크에서 사설 네트워크에 연결된 단말 호스트에 직접적인 통신이 불가능한 문제를 가진다. 이런 이유에서 분류 대상을 플로우로 선정하여 공유기의 설치 유무에 관계없이 단말 호스트의 네트워크 접근 방법이 유·무선인지를 분류하는 방법론을 제시하고자 한다.

2 장은 연구를 위해 구축된 환경과 실험 방법에 대하여 기술하였으며 3 장은 분석 방법론을 제안하고 있다. 4 장은 이를 바탕으로 구축된 호스트 정보 분석 시스템에 관하여 서술한다. 5 장은 호스트 정보 분석 시스템을 이용하여 얻어진 실험 결과를 기술한다. 마지막으로 6 장은 전체적인 결론 및 향후 연구과제에 대하여 기술한다.

이 논문은 2007 년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임 (KRF-2007-331-D00387)

2. 실험 방법

Libpcap[1]은 트래픽 분석 및 패킷 분석 연구에서 가장 널리 사용되는 라이브러리로 본 연구에서도 libpcap 기반의 tcpdump[2]를 이용하여 덤프파일을 작성한다. 실험환경은 학내 망으로 선정하였다. 실험 환경 구성은 크게 사설 네트워크와 공인 네트워크로 구분하였다. 사설 네트워크는 embedded linux 기반의 DD-WRT[3]를 이용하여 AP 구성 및 덤프파일을 작성하였다. 공인 네트워크 구간은 학내 AP 를 이용하였으며 Core 스위치의 미러링 구성을 통하여 덤프파일을 작성하였다. 이렇게 수집 시스템으로 두 곳을 결정한 이유는 본 실험의 결과를 통해 hop 이 증가하더라도 제안된 방법론이 유효함을 증명하기 위함이다.

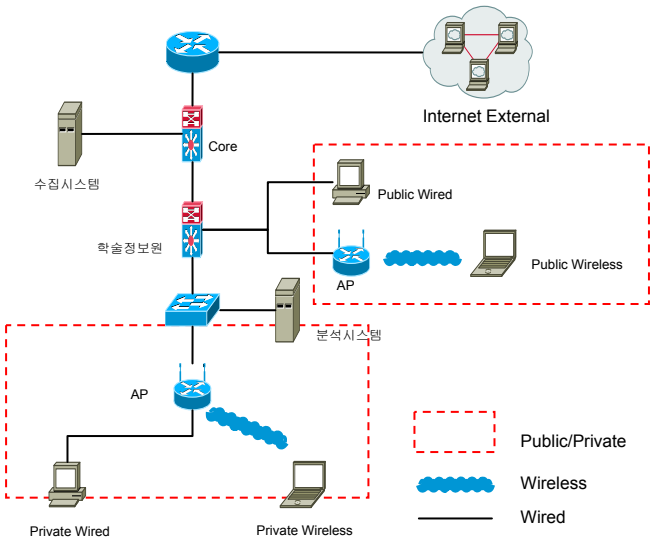
서버와 클라이언트의 운영체제에 대한 대조군 선택은 표 1 과 같다.

<표 1> 운영체제 대조군 목록

서버	클라이언트
Windows 2003 Server Standard	Windows XP SP3 Windows Vista SP1
Linux CentOS v5.2	Linux Ubuntu Solaris 10

이는 운영체제에 따른 트래픽이 상이하게 생성됨을 보이기 위함이다.

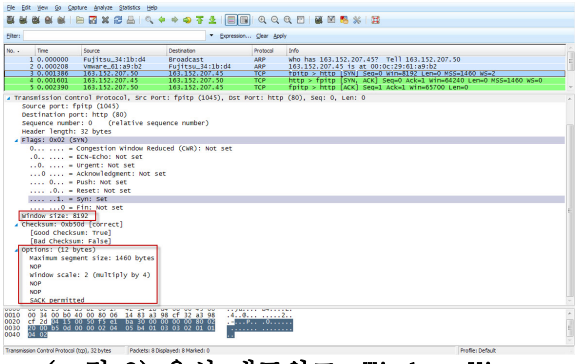
다음의 그림 1 은 실험환경의 전체적인 구성도를 나타낸 것이다. 점선은 각각 공인/사설 네트워크 구간을 나타낸다.



(그림 1) 실험 환경 구성도

3. 분류 방법

TCP 는 일반적인 네트워크에서 가장 많이 사용되는 프로토콜이다. 3-way handshaking 과정을 통해 연결 설정 후 C-S(Client-Server)간의 통신이 이루어 진다. 이 과정에서 SYN, SYN/ACK 에서 TCP option field 와 window size 의 값이 운영체제에 의해 결정된다. 예를 들어 Windows Vista 의 경우 TCP 의 SYN 패킷에 관한 window size 값은 8,192 이며 TCP option field 의 구성은 MTU 값으로 1,460 을 가지고, window scale 옵션이 활성화되어 인자값으로 2 를 취한다. 또한 SACK Permitted 옵션이 활성화된다. 이를 그림 2 에서 확인할 수 있다.



(그림 2) 유선 네트워크, Windows Vista

실험을 통하여 동일 조건에서 호스트의 운영체제가 변경됨에 따라 window size 와 TCP option field 의 값이 다양하게 변화된다는 것을 확인할 수 있다. 운영체제별 TCP 의 SYN 패킷 헤더 정보는 표 2 에 정리하였다. 표 2 의 유·무선 항목은 유·무선 트래픽 분류 가능 여부를 나타내고, 소괄호는 무선 트래픽의 경우 해당 값을 표기한 것이다. 표 2 을 기반으로 하여 운영체제를 예측하는 기준으로 사용할 수 있다. 본 논문에서도 표 2 의 내용을 기반으로 한 운영체제 분류를 한다.

<표 2> 운영체제별 SYN 패킷 헤더 정보

운영체제	Window Size	TCP Options	Value	유·무선
Windows XP	65,535 (16,384)	MSS	1,460	O
		SACK Permitted	·	
Windows Vista	8,192	MSS	1,460	X
		Window Scale Factor	2	
		SACK Permitted	·	
Linux	5,840	MSS	1,460	X
		Timestamp	·	
		Window Scale Factor	7	
		SACK Permitted	·	
Solaris	49,640	MSS	1,460	X
		Window Scale Factor	0	
		SACK Permitted	·	

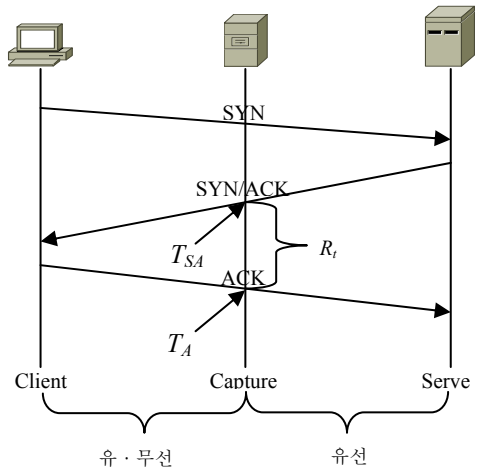
3.1. TCP Options

TCP 의 option field 는 최대 40 바이트의 옵션 정보를 가질수 있으며 이러한 옵션 정보는 단일 바이트와 다중 바이트 옵션의 범주로 나누어 질수 있다. 첫 번째로 단일 바이트로 이루워진 옵션에는 옵션의 끝을 의미하는 EoP(End of Option)와 무동작을 의미하는 NOP(No Operation)가 있으며, 다중 바이트로 이루워진 최대 세그먼트 크기(Maximum Segment Size), 윈도우 확장 인자(Window Scale Factor), 타임 스탬프(timestamp), 그리고 SACK 허용(SACK Permitted) 옵션이 있다. 이 중에서 TCP window size scale option 은 운영체제에 종속적인 사항으로 Windows Vista 와 Unix 계열, 리눅스는 기본적으로 활성화되어 있는 옵션이다. 특정 어플리케이션의 설치에 의해서 Windows XP 도 활성화가 되는 경우가 있다. 이 속성은 window size 를 증가시키기 위해 사용된다. 일반적인 window size 값인 65,535 는 window Size 로서는 매우 큰 값처럼 보이지만, 고속의 처리율과 긴 지연시간을 가진 전송 매체를 통해서 전달될 때에는 이 크기가 충분하지 않을 수 있다. 새로운 window size 를 구하기 위하여, 먼저 window scale factor 에 명시된 값만큼 2 의 급수를 구한다. 그 이후 이 결과를 헤더에 있는 window size 의 값에 곱함으로 새로운 window size 를 결정한다. Window scale option 이 활성화된 경우 인자값은 운영체제별로 차이가 존재하지만, 유·무선 트래픽에서는 동일한 window size 와 window scale factor 값을 가지게 된다. 따라서 window scale factor 와 window size 만을 가지고 유·무선 트래픽을 분류하는 기준으로는 사용할 수 없다. 단, 운영체제가 Windows XP 인 경우 window size 만을 이용하여 유·무선 트래픽을 분류할 수 있다. 이는 표 2 을 통해서도 확인이 가능하다.

3.2. 응답시간

무선 네트워크의 경우는 클라이언트가 AP 를 통하여 통신하기 위하여 경쟁을 하게 된다. 이는 클라이언트의 수가 증가함에 따라 심해지게 되며 이러한 이유로 지연이 발생하게 된다. 무선 네트워크의 보안을 강화하기 위하여 암호화를 설정한 경우도 처리에 따른 지연이 발생한다. 클라이언트와 AP 간의 장애물

유무에 따라서도 지연이 발생하며, 이 밖의 다양한 지연 문제로 유선에 비해 응답시간에 대한 값이 무선 네트워크에서는 커지게 된다. 이를 이용하여 유·무선 트래픽 분류의 기준으로 활용할 수 있다.



(그림 3) 응답시간

본 논문에서 응답시간에 대한 기준은 다음과 같이 정의하였다.

그림 3 에서 보이는 바와 같이 클라이언트는 서버와 TCP 연결을 하기위해 SYN 패킷을 전송하게 된다. 이에 대한 응답 신호로 서버는 클라이언트에게 SYN/ACK 패킷을 전송하게 된다. 마지막으로 클라이언트가 응답 신호의 의미로 서버에게 ACK 패킷을 전송한다. 수집 시스템은 서버와 클라이언트 사이에서 동작하게 된다. 클라이언트와 수집 시스템 사이는 유·무선 네트워크로 연결되며 수집 시스템과 서버사이는 유선으로 연결된다. 이 과정에서 유·무선이 존재하는 구간의 응답시간만을 구하기 위하여 다음과 같은 계산식이 사용된다.

$$R_t = T_A - T_{SA}$$

R_t 은 응답시간이며, T_A 는 ACK 패킷의 수집된 시간 T_{SA} 는 SYN/ACK 패킷의 수집된 시간을 의미한다. R_t 은 각 호스트의 플로우를 대상으로 계산이 이루어 진다.

응답시간에 따른 분포를 통해 집합을 생성하게 되고, 집합의 기준은 표준편차를 이용하게 된다.

$$Aver(R_t) = \frac{1}{N_f}\sum R_t$$

$$Diff(R_t) = R_t - Aver(R_t)$$

$$\sigma = \sqrt{\frac{1}{N_f}\sum \{Diff(R_t)\}^2}$$

위의 식에서 R_t 는 응답시간, N_f 는 플로우의 총 갯수, $Aver(R_t)$ 은 응답시간의 평균을 의미한다. $Diff(R_t)$ 는 응답시간에 대한 편차, σ 은 표준 편차를 나타낸다. 집합을 결정해주는 식은 아래와 같다.

If $Diff(R_t) - \sigma \geq 0$

then Group_A
else Group_B

Group_A 는 무선 호스트의 집합이 되며, Group_B 는 유선 호스트의 집합이 된다.

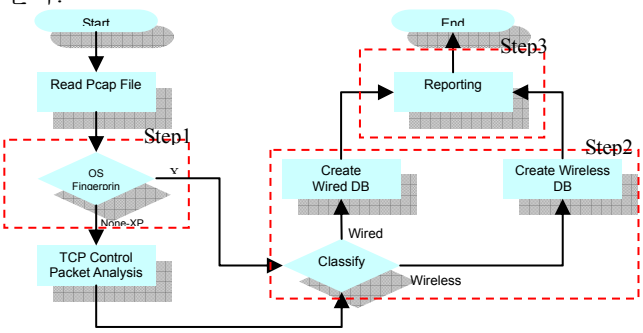
3.3. 분석 방법

호스트 정보 분석 시스템은 그림 4 와 같은 흐름으로 크게 3 단계로 진행한다. 3 단계 모듈의 구성은 OS Fingerprinting, Classification Traffic, Reporting 으로 구성되며, 최초 입력값으로는 libpcap 기반의 pcap 덤프파일을 사용한다. 덤프파일은 TCP 의 3-way handshake 만을 작성한다. 3-way handshake 만을 수집하기 위하여 수집 시스템에서는 다시 한번 분류 작업을 하게 되며, 또한 환경 조건을 유사하게 만들기 위하여 웹 서비스에 대한 패킷만을 대상으로 하였다.

덤프파일의 패킷을 입력받아 passive TCP finger-printing 을 하게 된다. 이 과정은 호스트의 운영체제가 Windows XP 인 경우 트래픽 분류를 위한 처리량을 줄일수 있다. 운영체제를 판별하는 과정은 다음과 같다.

i) TCP SYN 패킷의 option field 확인
ii) window size 확인
iii) 운영체제 판별

운영체제 판별은 TCP option field 의 인위적인 변경사항이 없는 호스트만을 선택하기위하여 None 이라는 분류 범위를 만들었다. 또한 판별기준은 표 2 을 사용한다.



(그림 4) 분석 시스템 흐름도

출력값은 플로우 정보를 효율적으로 생성, 저장하기 위하여 데이타베이스를 사용한다. 테이블의 구성은 표 2 와 같다.

<표 3> Packet 테이블 구성

필드	타입	Null	Key	Extra
No	int	No	Pri	Auto_increment
Hash	bigint	No		
P_S	int	No		
Pkt	varchar	No		
Seq	bigint	Yes		
Ack	bigint	Yes		
Flag	varchar	Yes		

표 3 의 해쉬 키 값은 source IP, destination IP, source port, destination port, flag, sequence number, acknowledgement number 를 이용하여 계산된다. 중복된 해쉬 키 값에 대한 충돌을 해결하기 위해서 다음과 같은 방법을 사용한다.

생성된 해쉬 키 값이 데이타베이스의 테이블 내에

존재하지 않으면 부모로 생각하며 P_S(Parent-Child) 필드에 '0'으로 표기한다. 만약 해쉬 키 값이 존재하면 최후에 추가된 데이터의 인덱스 번호를 P_S 필드의 값으로 가지게 함으로써 충돌을 해결할 수 있다.

Classification Traffic 은 데이타베이스의 패킷 테이블을 입력값으로 받아 SYN/ACK 패킷을 찾게한다. 이 과정에서 sequence number 와 acknowledgement number 를 이용하여 해쉬 키 값을 계산한 후 해당 플로우를 검색하게 된다. 검색된 데이타를 이용하여 응답시간을 구한다. 이 결과값을 새로운 테이블에 저장하게 되며, 저장되는 테이블은 플로우 테이블이다.

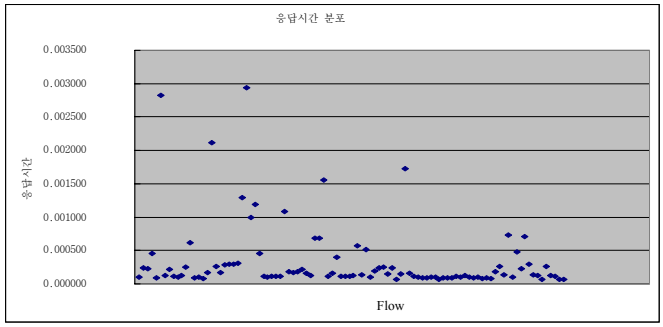
3.3. Report Result

Report Result 는 결과를 데이타베이스에 저장하여 웹을 통해 확인할 수 있도록 하기 위한 과정이다. 1 일을 기준으로 생성된 결과를 하나의 테이블로 구성하며, 필드는 IP, OS, 분류결과, 검증결과, 날짜로 구성된다.

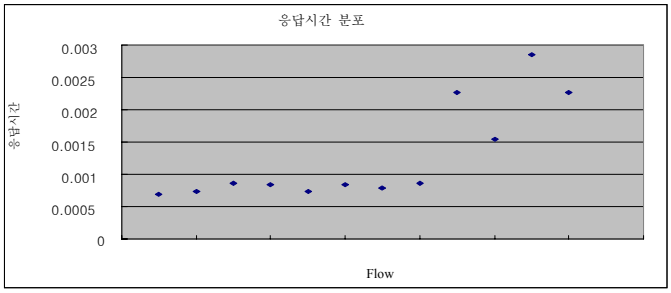
4. 실험 결과

학내 망의 트래픽은 분당 약 900MB 정도를 유지하였으며, 3 일간의 데이타를 이용하였다.

그림 5 와 그림 6 은 각각 공인/사설 네트워크 구간에서 수집된 호스트들의 응답시간에 대한 분포도를 나타낸다.



(그림 5) 공인 네트워크 구간의 응답시간 분포도



(그림 6) 사설 네트워크 구간의 응답시간 분포도

수집된 대상으로 분석 시스템에 적용한 결과, 정확한 플로우를 생성한 후 운영체제를 판별하였다. 이후 유·무선 트래픽을 정확하게 분류하였다. 또한 공인 네트워크와 사설 네트워크에서도 문제없이 판별이 가능하였다. 이는 앞서 기술한 hop 이 판별 기준에 영향을 미치지 않고 분류가 가능하다는 것을 보여준다. 유·무선 트래픽 분류에 대한 검증 방법은 공인 IP 를 DHCP 로부터 할당받은 호스트에 한해서 시행하였으며, 해당 네트워크 POOL 을 기준으로 검사하였다. 예를 들어 무선 네트워크로 할당해 주는 POOL 이 123.1.1.0/24 이며, 호스트의 IP 가 해당 POOL 에 속한다면, 이를 무선 호스트에 대한 정답으로 사용하였다. 운영체제 판별에 대한 정답지는 미리 운영체제가 조사된 호스트만을 대상으로 정확도를 측정하였다.

<표 4> 실험 결과

	공인 네트워크	사설 네트워크
운영체제 분류	100%	100%
유·무선 분류	98.2%	100%

표 4 는 공인 네트워크와 사설 네트워크에서의 실험 결과를 나타낸다.

공인 네트워크 구간에서의 정확도가 떨어지는 이유는 Windows Vista 가 운영체제인 경우였으며, 원인은 무선 네트워크 통해 트래픽을 발생하였을 경우, 동일한 플로우를 2 번 발생시키는 경우가 종종 발생하였다. 이는 2 번째 플로우가 작은 응답시간을 갖게 되어 유선 네트워크로 오분류하게 된다. 이러한 이유로 해당 호스트를 공유기로 잘못 판단하는 경우도 나왔으며, 이에 대한 원인을 규명하지 못하였다. 향후 연구를 통해 풀어야할 과제이다.

5. 결론 및 향후 연구 방향

선행 연구들에서는 ICMP 에 대한 응답을 기준으로 사용하였다. 이는 공유기가 설치된 IP 에 대하여서는 분석이 불가능하며, ICMP 를 방화벽에서 차단한 경우도 분석이 불가능하다. 그러나 본 논문에서는 호스트에서 발생시킨 트래픽을 분석한 후 응답시간 차에 따른 집합화를 이용하여 유·무선 트래픽을 분류한다. 이러한 방식은 선행 연구들의 단점을 극복할 수 있으며, 이를 이용한 유·무선 공유기 설치 유무 판별도 가능하다. 더욱이 본 논문에서 제시한 방법론은 passive 방식으로 전체 네트워크에 부하를 주지 않기 때문에 선행 연구에 의한 방법론보다 효율성이 뛰어나다.

향후 호스트의 운영체제의 버젼 정보까지 파악한다면 이를 기반으로 취약한 호스트를 파악할 수 있다. 이는 취약 서비스에 대한 패치가 이루어 질 수 있도록 하는 시스템 개발에 기반을 제공할 수 있다고 생각한다. 또한 TTL 값을 이용한 공유기 설치 유무 판별 알고리즘에 관한 연구도 계속 진행할 계획이다.

참고문헌

[1] Libpcap, http://www.tcpdump.org/pcap3_man.html.
[2] tcpdump, http://www.tcpdump.org/tcpdump_man.html.
[3] dd-wrt, http://www.dd-wrt.com/dd-wrtv3/index.php.
[4] Behrouz A. Forouzan, "TCP/IP Protocol Suit Third Edition", McGraw-Hill, 2005.
[5] p0f, http://lcamtuf.coredump.cx/p0f.shtml.
[6] wireshark, http://www.wireshark.org/.
[7] scapy, http://www.secdev.org/projects/scapy/.
[8] nmap, http://nmap.org/book/osdetect.html.