패킷 헤더 정보를 이용한 침입 유형 판별에 관한 연구

조 혁*, 김익수*, 김명호*
*숭실대학교 컴퓨터학과
e-mail : laphael7@ss.ssu.ac.kr

A Study on Intrusion Pattern Distinction using Packet Header Information

Hyuck Jo*, Ik-Su Kim*, Myung-Ho Kim*
*Dept. of Computing, Soongsil University

요 약

최근 여러 종류의 네트워크 공격들이 기업이나 연구소, 학교 심지어는 가정에까지 심각한 위협을 주고 있다. 이러한 공격을 침입 탐지 시스템으로 탐지하고 있지만, 기존의 침입 탐지 시스템은 본질적으로 패킷 수집의 능력이 떨어질 뿐만 아니라 과도한 패킷이 유입될 때 제 기능을 발휘하지 못하게 된다. 본 논문에서는 이러한 침입 탐지 시스템의 성능에 대한 문제점을 개선하기 위해 패킷 헤더 정보를 이용하여 시스템의 부하를 줄이고, 어떤 공격이 들어오는지를 파악할 수 있는 시스템을 제안한다. 본 논문을 통해서 각각의 공격툴을 탐지하는 기법과 알려지지 않은 공격에 대한 탐지룰을 생성하는 연구에 많은 도움을 줄 것으로 예상한다.

1. 서 론

최근들어 네트워크 공격에 대응하기 위한 여러 가지 침입탐지 시스템들이 연구되고 있다. 침입탐지 시스템(Intrusion Detection System:IDS)은 호스트기반의 침입탐지와 네트워크기반의 침입탐지로 나눌 수 있다. 호스트기반 IDS는 시스템의 로그 정보와 특정 행위에 대한 감사 자료 분석을 통해 침입을 탐지하며, 네트워크기반 IDS는 네트워크상의 패킷 정보를 분석하여 침입을 탐지한다. 특히 네트워크기반 침입탐지 시스템은 패킷의 모든 내용을 분석하기 때문에 네트워크 장비나 전체 시스템에 많은 과부하가 걸리므로 침입 탐지에 관한 성능 저하가 있을 수 있다. 특히 작은 크기의 패킷이 대량으로 들어오게 되면 패킷 삭제가 일어나 침입의 패턴을 잃어버리는 치명적인 결과를 낳게 된다[1,2,3].

본 논문에서는 패킷 헤더의 정보만을 분석하여 시스템 부하를 최소화 하고 공격 유형을 판별한 후 공격 패턴을 정의하는 방법을 제안한다. 이렇게 정의된 패턴은 침입탐지 시스템의 중요한 정보로 활용할 수 있을 것으로 기대한다. 본 논문의 구성은 2장에서 네트워크 공격툴들을 정리하고, 3장에서는 패킷 헤더 정보 분석에 필요한 시스템을 설명한다. 4장에서는 패킷 헤더 분석과 각각의 공격들에 대한 패턴을 정의하고, 마지막으로 5장에서는 결론 및 앞으로의 연구방향에 대해 언급한다.

2. 관련연구

이 장에서는 네트워크 공격유형들과 각 공격툴들의 특징을 알아본다.

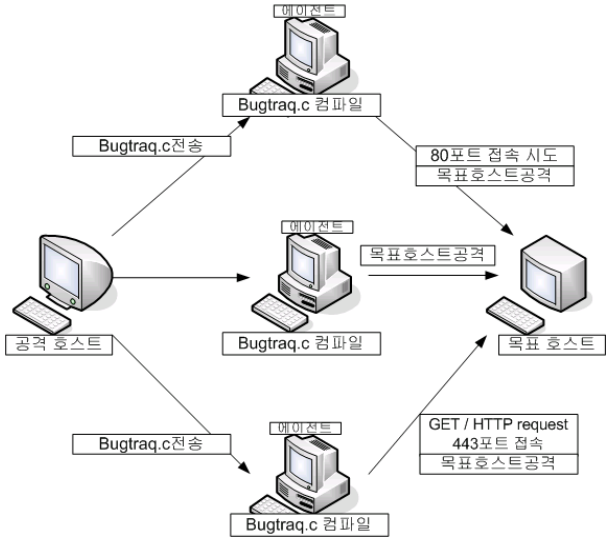
2.1 분산 서비스 거부 공격

분산 서비스 거부 공격이란 많은 수의 서비스 거부 공격용 프로그램들이 분산 설치되어 이들이 서로 통합된 형태로 목표 시스템에 일제히 데이터 패킷을 범람시켜 성능저하 및 시스템 마비를 일으키는 기법을 말한다. 이러한 분산 서비스 거부 공격 도구로는 Trinoo와 TFN, TFN2K, Stacheldraht가 대표적이며, 공격 형태로는 SYNflood 공격과 ICMPflood 공격, UDPflood 공격이 대표적으로 있다. SYNflood 공격은 TCP의 3-way handshaking 기법에 의한 연결 과정

의 취약성을 이용한다. ICMPflood공격은 직접 목표 호스트로 대량의 ICMP패킷을 발송한다. UDPflood 공격은 공격 대상을 향해 계속적인 UDP패킷을 보냄으로써 공격 대상 네트워크의 대역폭을 소모시켜 서비스를 불가능하게 만든다[4,5].

2.2 리눅스 웜

리눅스 웜의 한 종류인 Linux.slapper.worm은 아파치에서 클라이언트와 웹서버 사이에 안전한 통신을 보장하기 위한 방법으로 사용되는 OpenSSL(Socket Secure Layer)의 버퍼오버플로우 취약점을 이용하여 공격하며, 지정된 IP 대역에서 취약점을 가지고 있는 웹 서버를 찾아 감염시켜 웜을 확산 시킨다[6].



(그림 1) 리눅스 웜의 공격 구성도

(그림 1)을 보면, 공격호스트가 목표호스트의 웹서버 포트인 80번에 접속하여 “GET / HTTP” 요청을 하고 아파치 서버를 찾는다. 웹서버의 응답에 아파치 서버가 확인되면 포트 443번에 접속을 시도 하고, 성공적으로 접속이 되면 SSL의 취약점을 이용하여 공격을 시도하기 위한 통신을 하게 된다. 공격에 성공하게 되면 에이전트 컴퓨터로 “/tmp/.bugtraq.c” 소스파일을 전송하고, 이것을 에이전트가 컴파일 한 후 생성된 “/tmp/a” 파일을 실행시켜 자기 자신이 웜에 감염된다. 웜에 감염되면 해당 시스템에 백도어를 남기게 되고, 목표 호스트를 향해 UDPflood, TCPflood같은 분산 서비스 거부 공격을 하게 된다.

2.3 스캔 공격

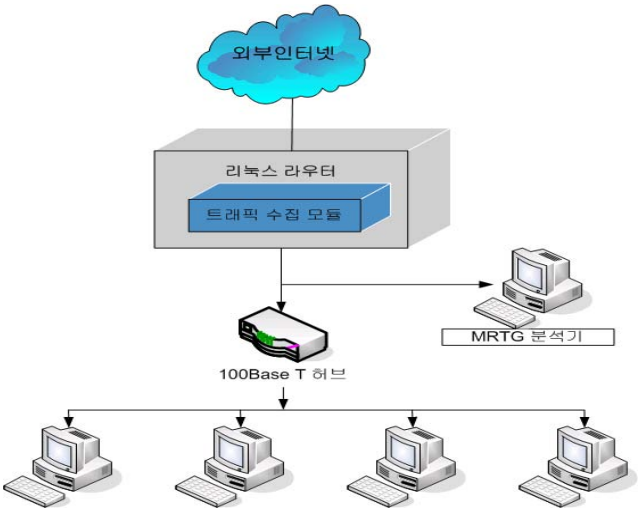
스캐닝이란 해킹을 위한 사전 단계로 목표호스트에 대한 정보 수집을 하는 행위이며, 크게 호스트 스캐닝과 포트 스캐닝으로 나뉘어 진다. 호스트 스캐닝은 특정 서비스에 대해서 취약성을 갖고 있는 호스트를 찾기 위해 공격자가 특정 포트에 대해 스캔을 하는 방식이며, 포트 스캐닝은 특정 호스트를 대상으로 하여 해당 호스트에서 제공하는 특정 서비스를 찾기 위해 포트번호를 바꾸어 스캔하는 방식이다. 스캔공격은 스캔을 통해 찾아낸 호스트의 포트를 이용하여 웜의 전파 및 백도어 등의 해킹 툴을 감염시키는 것이다.

3. 패킷 헤더 정보 수집을 위한 시스템 구현

이 장에서는 패킷 헤더 정보 수집을 위한 시스템 구조와 모듈에 대해서 살펴본다.

3.1 패킷 헤더 정보 수집을 위한 시스템 구조

패킷 헤더에는 프로토콜과 IP주소, 포트번호, 데이터 크기 같은 정보들이 있다. 이러한 정보를 수집하기 위해서 (그림 2)와 같은 시스템을 구현한다.



(그림 2) 패킷 헤더 정보 수집을 위한 시스템 구조도

(그림 2)에서 보듯이 패킷 헤더 정보 수집을 위한 시스템 구조는 외부에서 유입되는 패킷 헤더 정보를 수집하기 위해 트래픽 수집 모듈을 설치한 리눅스 라우터와 데이터의 유입량을 알아보기 쉽도록 설치한 MRTG(Multi Router Traffic Grapher) 분석기로 구성된다. 리눅스 라우터는 저비용으로 구축되며 강력한 라우터 기능을 수행한다. 라우터에 트래픽 수집 모듈을 설치한 이유는 공격자의 침입을 탐지하기 위해서는 네트워크를 통해 유입되는 모든 트래픽을 감시해야 하기 때문이다.

3.2 패킷 헤더 정보 수집을 위한 모듈 작성

패킷 헤더 정보 수집을 위한 모듈은 PCAP(Packet Capture library)를 사용하여 네트워크에 들어오는 패킷 헤더 정보를 수집한다. PCAP은 사용자 수준에서 시스템에 상관없이 패킷 수집을 용이 하도록 도와주는 라이브러리다[7]. PCAP을 이용하여 구현된 트래픽 수집 모듈은 네트워크 트래픽으로부터 필요한 헤더 정보를 추출하여 저장하게 된다

```
ICMP (소스 IP 주소) (목적지 IP 주소) (15초간 누적 패킷 길이) (15초간 누적 패킷 수)
TCP (소스 IP 주소) (목적지 IP 주소) (소스 포트번호) (목적지 포트번호) (15초간 누적 패킷 길이) (15초간 누적 패킷 수)
UDP (소스 IP 주소) (목적지 IP 주소) (소스 포트번호) (목적지 포트번호) (15초간 누적 패킷 길이) (15초간 누적 패킷 수)
```

(그림 3) 모듈을 통해 생성된 헤더 정보의 포맷

(그림 3)은 트래픽 수집 모듈을 통해 생성된 트래픽 헤더 정보에 대한 포맷이다. 첫 번째 필드는 해당 패킷의 프로토콜을 나타내며 TCP, UDP 패킷의 경우에는 소스/목적지 IP 주소와 소스/목적지 포트번호, 15초간 누적된 패킷들의 총 길이, 15초간 누적된 패킷 수를 나타낸다. ICMP 패킷의 경우에는 소스/목적지 포트번호가 없다는 것을 제외하고 TCP, UDP 패킷과 동일한 정보를 가진다.

4. 공격 프로그램별 유형 분석

공격 유형 분석은 분산 서비스 거부 공격과 리눅스 웜, 스캔도구로 공격을 수행한 후, 패킷 캡쳐 모듈을 사용하여 트래픽 정보를 수집 분석 하였다.

4.1 분산 서비스 거부 공격 분석

분산 서비스 거부 공격은 Trinoo와 TFN의 공격방법과 유사한 Stacheldraht를 사용하여 다양한 공격을 실시하였다.

protocol	source IP	destonation IP	s-port	d-port	size	패킷수
(Stacheldraht_UDP-bomb)						
UDP	xxx.xxx.xxx.215	xxx.xxx.xxx.79	12512	12512	1052	1
UDP	xxx.xxx.xxx.23	xxx.xxx.xxx.79	18929	18929	1052	1
(Stacheldraht_SYN-flooding)						
TCP	xxx.xxx.xxx.32	xxx.xxx.xxx.79	1147	60	40	1
TCP	xxx.xxx.xxx.58	xxx.xxx.xxx.79	1147	59	40	1
(Stachceldraht_ICMP-flooding)						
ICMP	xxx.xxx.xxx.21	xxx.xxx.xxx.79			1072	1
ICMP	xxx.xxx.xxx.45	xxx.xxx.xxx.79			1072	1

(그림 4) Stacheldraht의 서비스 거부 공격 로그정보

(그림 4)는 Stacheldraht를 이용한 분산 서비스 거부 공격의 로그정보를 보여준다. 분산 서비스 거부 공격은 다음과 같이 3가지로 나누어 졌다. 첫 번째 UDP_flood 공격은 UDP패킷을 사용하며, 송신 IP만 변하는 것을 알 수 있었다. 송수신 포트는 모두 변하였지만 서로 응답을 주고받는 포트는 동일하였으며, 패킷 크기는 패킷 1개당 1052바이트로 일정하였다. 두 번째 SYNflood 공격은 TCP패킷을 사용하며, 송신 IP만 변하는 것을 알 수 있었다. 송신 포트는 공격이 시작 되면 고정이 되었고 수신포트는 임의의 포트로 변하였다. 패킷 크기는 패킷 1개당 40바이트로 일정하게 유지 되었다. 세 번째 ICMP_flood 공격은 ICMP패킷을 사용하며, 송신 IP만 변하는 것을 알 수 있었고 패킷 크기는 패킷 1개당 1072바이트로 일정하였다. 따라서 (그림 5)와 같이 공격패턴을 정의할 수 있다.

```
if use UDP and 송신 IP 변화 and 송수신 포트 동일
                패킷크기 1052바이트 = UDPflood 공격
if use TCP and 송신 IP 변화 and 송신 포트 고정, 수신포트 변동, 패킷크기 40바이트 = SYNflood 공격
if use ICMP and 송신 IP 변화 , 포트사용안함
                패킷크기 1072바이트 = ICMPflood 공격
```

(그림 5) Stacheldraht의 공격패턴 정의

4.2 웜공격

웜 공격은 Linux.slapper.Worm을 사용하여 공격을 실시하였다.

protocol	source IP	destonation IP	s-port	d-port	size	패킷수
(Linux.slapper.Worm)						
TCP	xxx.xxx.xxx.77	xxx.xxx.xxx.79	32788	80	300	5
TCP	xxx.xxx.xxx.76	xxx.xxx.xxx.79	33123	443	60	1

(그림 6) Linux.slapper.Worm공격 로그정보

(그림 6)은 Linux.slapper.Worm공격의 로그정보를 보여준다. Linux.slapper.Worm은 TCP 패킷을 사용하였으며, 각각의 에이전트에서 하나의 호스트를 공격하기 때문에 송신 IP는 변하게 되고, 수신 IP는 하나로 고정 되어 있는 것을 알 수 있었다. 공격이 시작되면 송신 포트는 변하고, 수신 포트는 80포트와 443포트로 고정되어 있었다. 패킷 크기는 패킷 1개당 60바이트로 일정하였다. (그림 7)은 Linux.slapper.Worm의 공격패턴을 정의한 모습이다.

```
if use TCP and 송신 IP 변화, 목적지 443포트사용
                패킷크기 60바이트 = 리눅스 웜 공격
```

(그림 7) Linux.slapper.Worm의 공격패턴 정의

4.3 스캔공격

스캔공격에는 다양한 종류의 공격방법이 있으며 이 실험에서는 3가지 스캐너로 실험을 하였다.

protocol	source IP	destonation IP	s-port	d-port	size	패킷수
(Stealth scanner)						
TCP	xxx.xxx.xxx.63	xxx.xxx.xxx.79	3230	80	270	5
TCP	xxx.xxx.xxx.63	xxx.xxx.xxx.79	3231	80	292	5
(Port scanner)						
TCP	xxx.xxx.xxx.63	xxx.xxx.xxx.79	3291	319	40	1
TCP	xxx.xxx.xxx.63	xxx.xxx.xxx.79	3292	320	80	2
(nmap scanner)						
TCP	xxx.xxx.xxx.63	xxx.xxx.xxx.79	58037	24	40	1
TCP	xxx.xxx.xxx.63	xxx.xxx.xxx.79	58037	342	40	1
TCP	xxx.xxx.xxx.63	xxx.xxx.xxx.79	58037	1312	40	1

(그림 8) 스캔 공격 로그정보

(그림 8)은 스캔 공격의 로그정보를 보여준다. 스텔스 스캐너는 TCP 패킷을 사용하며 송수신 IP가 고정되는 것을 알 수 있었다. 송신 포트는 하나씩 증가하고, 수신 포트는 80포트로 고정되어 있었다. 목표 호스트의 취약성 파일을 검색하는 특징 때문에 파일의 크기가 일정하지 않고 계속 변하였다. 포트 스캐너는 TCP 패킷을 사용하며, 송수신 IP가 고정되는 것을 알 수 있었다. 송신 포트는 포트번호 1번부터 65532번 포트까지 모두 검색하는 포트스캐너의 특성상 포트가 하나씩 증가 하였다. 파일의 크기는 패킷 1개당 40바이트로 일정하게 유지되었다. nmap스캐너는 TCP패킷을 사용하며, 송수신 IP는 고정되는 것을 알 수 있었다. 특히 다른 스캐너와 달리 송신 포트가 한번 결정되면 변하지 않고 고정되어 있었다. 파일의 크기는 패킷 1개당 40바이트로 일정하게 유지되었다. (그림 9)는 스캔공격의 패턴을 정의한 모습이다.

if use ***TCP*** and ***송수신 IP 고정, 목적지 포트 80포트 고정***
패킷크기 계속변동 = 스텔스 스캐너 공격
if use ***TCP*** and ***송수신 IP 고정, 목적지 포트 1씩 증가***
패킷크기 ***60***바이트 = ***포트 스캐너*** 공격
if use ***TCP*** and ***송수신 IP 고정, 송신포트고정***
패킷크기 ***40***바이트 = ***nmap 스캐너*** 공격

(그림 9) 스캔 공격의 패턴 정의

패킷 헤더 정보 분석 결과 일반적으로 사용하는 패킷 헤더 정보와는 달리 다수의 패킷이 일정한 크기로 짧은 시간 안에 라우터를 지나가면 네트워크 공격이라고 판단 할 수 있었고, 각각의 공격별로 일반적인 패킷 헤더 정보와 비교 했을때 확실한 차이를 보였다. (그림 10)은 일반적인 패킷 헤더의 로그정보를 나타낸 것으로 ssh접속과 web접속, ftp접속을 나타낸다. 이러한 분석결과를 토대로 각각의 공격마다 일정한 패턴을 정의 하여 침입 탐지 시스템에서 사용하는 룰을 제공할 수 있다.

protocol	source IP	destonation IP	s-port	d-port	size	패킷수
ssh:TCP	xxx.xxx.xxx.63	xxx.xxx.xxx.79	2206	22	40	1
web:TCP	xxx.xxx.xxx.53	xxx.xxx.xxx.79	45119	80	1315	5
ftp:TCP	xxx.xxx.xxx.35	xxx.xxx.xxx.79	21	32831	264	3

(그림 10) 일반적인 패킷 로그정보

5. 결론 및 향후 과제

본 논문에서는 네트워크상에서 심각한 문제를 일으키는 분산 서비스 거부 공격과 리눅스 웜, 스캔공격을 효율적으로 탐지하기 위한 방법으로 패킷 헤더 정보를 분석하여 공격 패턴을 정의하였다. 공격 패턴은 패킷 헤더 정보에 있는 프로토콜과 IP주소, 포트번호, 데이터의 크기를 분석하여 정의하였다. 이렇게 정의된 패턴을 침입 탐지 시스템에서 사용한다면 기존의 침입 탐지 시스템에서 가지고 있던 시스템 부하의 문제점과 패킷을 잃어버리는 문제점들을 해결 할 수 있을 뿐만 아니라 알려지지 않은 공격을 탐지하기 위한 학습시스템을 구축할 시 중요한 핵심 정보로 사용될 수 있다.

앞으로의 연구 과제는 알려지지 않은 네트워크 공격의 침입 패턴생성 자동화와 이러한 침입 패턴들을 가지고 실제 침입을 판별 할 수 있는 모듈을 만드는 과제가 남아 있다.

참 고 문 헌

[1] James Cannady, Jay Harrell, "A Comparative Analysis of Current Intrusion Detection Technologies," February 1998.

[2] Mansour Esmaili, Rei Safavi-Naini, "Case-Based Reasoning for Intrusion Detection," Computer Security Applications Conference pp.214-222, 1996.

[3] Kevin J. Houle, George M. Weaver, "Trends in denial of Service Attack Technology," CERT Coordination Center, October 2001.

[4] Frank Kargl, Joern Maier, Michael Weber, "Protecting Web Servers from Distributed Denial of Service Attacks," May 2001.

[5] L. Garber, "Denial-of-Service Attack Rip the Internet", Computer, April 2000.

[6] http://www.trendmicro.com/search/google/en-us/bugtraq

[7] http://kmh.ync.ac.kr/Hacking/2nd/node203/