IP Fragmentation 공격 탐지를 위한 실시간 접근 로그 설계 및 구현

국 경 완† · 이 상 훈††

요 약

네트워크가 보편화되면서 사이버 공간을 이용한 테러가 전 세계적으로 발생하고 있다. IP Fragmentation은 이 기종 네트워크 환경에서 IP 패킷의 효율적인 전송을 보장해주고 있지만, 몇 가지 보안 문제점을 가지고 있다. 불법 침입자는 이러한 IP Fragmentation의 취약점을 이용해 IP Spoofing, Ping of Death, ICMP 등의 공격 기술을 이용하여 시스템에 불법적으로 침입하거나 시스템의 정상적인 동작을 방해한다. 최근에는 IP Fragmentation을 이용한 서비스 거부공격 외에도 이를 이용하여 패킷 필터링 장비나 네트워크 기반의 침입탐지시스템을 우회할 수 있는 문제점이 대두되고 있다. 본 논문에서는 패킷 재 조합 기능을 제공하지 못하는 일부 라우터나 네트워크 기반의 침입탐지시스템들에서 불법 사용자가 패킷을 다수의 데이터 그램으로 나누어 공격할 때, 이를 탐지하거나 차단하지 못하는 경우에 대비하여 실시간 접근 로그 파일을 생성하고, 시스템 관리자가 의사결정을 할 수 있도록 함과 동시에 시스템 스스로 대처할 수 있는 시스템을 구현하여 타당성을 검증하고, 그에 따른 기대 효과를 제시하고자 한다.

Design and Implementation of a Real Time Access Log for IP Fragmentation Attack Detection

Kyoung-Wan Kug† · Sang-Hoon Lee††

ABSTRACT

With the general use of network, cyber terror rages throughout the world. However, IP Fragmentation isn't free from its security problem yet, even though it guarantees effective transmission of the IP package in its network environment. Illegal invasion could happen or disturb operation of the system by using attack mechanism such as IP Spoofing, Ping of Death, or ICMP taking advantage of defectiveness, if any, which IP Fragmentation needs improving. Recently, apart from service refusal attack using IP Fragmentation, there arises a problem that it is possible to detour packet filtering equipment or network-based attack detection system using IP Fragmentation. In the paper, we generate the real time access log file to make the system manager help decision support and to make the system manage itself in case that some routers or network-based attack detection systems without packet reassembling function could not detect or suspend illegal invasion with divided datagrams of the packet. Through the implementation of the self-managing system we verify its validity and show its future effect.

키워드 : IP Fragmentation, 실시간 접근 로그 설계(Real time access log)

1. 서 론

연구와 군사 목적으로 시작된 인터넷은 현재 많은 기업들과 일반인들에게 선택이 아닌 필수의 도구가 되었고, 현대 사회에서 없어서는 안 될 중요한 기반으로 자리 잡게 되었다. 더불어 보안문제가 인터넷 사회에 끼치는 영향이 커짐에 따라 보다 많은 침입자들이 특정 목적을 위하여 보안모델의 취약점을 공격하고 있으며, 공격용 프로그램 또한 보다 복잡하고 정교해 지고 있다. 즉, 인터넷의 상용화 및 영향력이 커짐에 따라 침입자들이 자신들의 무기를 정교히 만들만한 동기를 갖게 된 것이다[11]. 과거에는 단순히 특정 시스템의 버그를 공격하는 도구 및 이러한 취약점을 찾아주는 스캔 공격도구들이 주류를 이루었으나, 현재에는 좀 더 나아가 방화벽시스템(Firewall System) 및 기타 보안시스템을 우회하기 위한 좀더 진보된 종류의 공격도구들이 나타나고 있다.

† 준 회 원 : 국방대학교 전산정보학과 연구원
†† 종신회원 : 국방대학교 전산정보학과 교수
논문접수 : 2001년 9월 25일, 심사완료 : 2001년 12월 24일

기존의 공격방법은 이미 잘 알려져 있어 침입차단시스템(IDS) 등을 이용하여 공격패턴에 대한 탐지가 가능하였고, 보안시스템 구축을 통하여 적절히 방어할 수 있었다. 이러한 보안기술의 발전과 더불어 이를 극복하기 위한 공격기술 또한 발전하였고 새로운 공격모델이 등장하였다. 새로운 공격모델에서는 보다 복잡한 공격 탐지 기법이 필요하며, 그 대응방법에 대한 변화를 요구하고 있다. 반면 공격자 입장에서는 공격도구의 자동화로 인하여 공격기술이 대중화되고 있으며, 따라서 네트워크 관련 공격이 점점 많아지고 있는 실정이다.

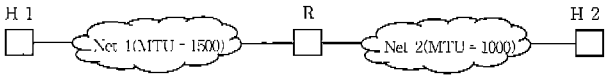
IP Fragmentation은 하나의 패킷(Packet)이 너무 커서 하나의 엔티티(entity)로서 전송되어질 수 없을 때, 네트워크를 통해 보내어질 수 있는 두 개 이상의 더 작은 패킷 조각(piece)으로 분리는 것을 말한다[8]. IP Fragmentation은 TCP/IP 상에서 매우 빈번하게 이루어지며, 이러한 기술은 네트워크 상에서 IP 패킷의 효율적인 전송을 보장해 주지만 몇 가지 문제점을 가지고 있다. 일부 라우터, 방화벽시스템이나 침입차단시스템은 패킷 재구성작업(packet reassembling)을 수행하지 않기 때문에 공격자들은 인공적으로 시스템의 기능장애를 발생시키거나, 방화벽의 보안정책을 우회시키고, 침입탐지시스템의 탐지 정책을 피하기 위하여 다수의 분할된 패킷들로 나누어 공격할 경우 이를 탐지하거나 차단하지 못하는 경우가 발생하고 있다[10]. 본 논문에서는 IP Fragmentation 개념과 tcpdump 프로그램을 이용하여 패킷을 모니터링 하는 방법과 IP Fragmentation의 취약점을 이용한 공격유형과 대처 방안에 대하여 살펴보고, 방화벽시스템, 침입탐지시스템 등과 같은 보안 도구들이 IP Fragmentation을 이용하여 공격했을 경우 이를 탐지하거나 차단하지 못할 경우를 대비하여 실시간 접근 로그를 생성하여 시스템을 보호할 수 있는 방법을 제시한다.

본 논문의 구성은 다음과 같다. 제2장에서는 IP Fragmentation의 개념과 네트워크상의 패킷을 모니터링 할 수 있는 tcpdump 프로그램을 이용하여 패킷이 분할(fragment)되는 과정에 대하여 살펴보고, 제3장에서는 이러한 IP Fragmentation의 보안 허점을 이용한 공격기술과 대처 방안에 대해 기술한다. 제4장에서는 IP Fragmentation이용하여 공격할 때 실시간 접근 로그를 생성하는 시스템을 설계 및 구현하고, 시스템을 검증하였다. 제5장에서는 본 논문의 결론을 맺고 향후 연구 방향에 대하여 기술하였다.

2. IP Fragmentation 개념과 패킷 모니터링

네트워크는 서로 다른 기종의 시스템으로 구성되어 있는 경우가 대부분이므로 동일한 형태의 프레임으로 자료를 전송 할 수 없다. 즉, 자료를 보내고자 하는 소스 호스트는 자료를 패킷이라고 하는 작은 블록으로 나누고 이를 목적지 주소와 함께 묶어 보낸다. 이때 각 패킷 각각의 조각을 fragment라 하며 시스템 및 네트워크 장비에서 이러한 패킷을 나누는 작업을 fragmentation이라 한다. fragmentation은 TCP에서 가장 빈하게 이루어지지만 UDP(User Datagram Protocol), ICMP(Internet Control Message Protocol) 프로토콜 등에서도 fragmentation이 이루어진다. 이와 같이 IP 프로토콜은 IP 패킷을 몇 개의 작은 패킷으로 나누어서 전송되고 목적지 시스템에서 재 조합되는 것이 허용되며, 서로 다른 최대 패킷 사이즈의 제한을 가진 이 기종의 전송매체에서도 IP 데이터 그램(Datagram)을 전송 가능하게 한다.

(그림 1)과 같이 호스트 H1에서 호스트 H2로 1500바이트 크기의 데이터 그램을 전송한다고 하면 라우터 R은 데이터 그램을 수신할 것이고, NET 2를 통해 전송할 수 없을 것이다. 왜냐하면 NET 2의 MTU(Maximum Transmission Unit) 용량이 1000 바이트 크기밖에 되지 않기 때문이다. 이 문제를 해결하기 위해 단편화란 기술을 이용하여 데이터 그램을 쪼개고 프레임으로 캡슐화(Encapsulation)하여 보내는 것이다[8].



(그림 1) IP Fragmentation

이처럼 fragmentation은 지극히 일반적이고 정상적인 이벤트이지만, 비정상적인 fragment를 발생시켜 서비스 거부 공격에 이용하기도 하고, fragmentation을 처리하지 않는 라우터나, 침입탐지시스템을 피하기 위한 목적에서 고의로 fragmentation을 이용하기도 한다. 목적지로 모든 fragment들과 데이터 그램이 안전하게 도착을 했다면 이를 재조립을 해야 하는데, 이를 중간 단계의 라우터에서 하지 않는 이유는 네트워크 트래픽을 줄이기 위해서이다. 라우터는 수신되는 데이터 그램이 fragment인지 아닌지를 알 필요가 없어지며, 수신 즉시 전달할 수 있다. 만약 라우터가 재조립을 한다면 모든 단편들을 다 수신할 때까지 기다려야 하기 때문에 네트워크 트래픽이 증가한다. 데이터 그램의 Identification에 fragment들의 식별번호를 저장하며 목적지 호스트는 이 정보를 참조하여 재조립을 한다. 각 fragment들은 목적지에 도착하여 fragment되기 전의 상태로 재 조합되기 위하여 다음의 정보들을 가지고 있다[10].

- 각 fragment는 하나의 동일한 fragment 식별번호를 이용하여 재 조합되는데, 이 식별번호는 IP 헤더의 16

비트 필드로써 "IP Identification Number" 또는 "Fragment ID"로 불린다.

- 각 fragment는 원래 fragment되기 이전의 패킷에서의 위치 즉 "Fragment Offset"을 가진다.
- 각 fragment는 그 fragment의 데이터 길이를 가진다. 여기서 IP 헤더 20byte는 데이터 길이에서 제외된다. 즉, Ethernet의 MTU인 1500바이트가 전송될 때 데이트길이는 1480(1500-20)바이트로 표시된다.
- 마지막으로 각 fragment는 현재 fragment에 추가적인 fragment들이 있을 경우 ME(More Fragment) flag를 1로 설정된다.

IP Fragment의 원리를 이해하기 위하여 4,000바이트의 ICMP 데이터가 Ethernet 상에서 전송될 때 어떻게 fragmentation되는지 살펴보기 위하여 (그림 2)와 같이 4000바이트의 ICMP 데이터를 송신한다. 일반적으로 사용되는 ping 패킷은 56바이트나 64 바이트의 ICMP 데이터를 전송하기 때문에 -s 옵션을 이용하여 ICMP 데이터가 fragment 되도록 충분히 크게 설정한다.

```
[root@kugstone /root]# ping -s 4000 192.168.0.1
PING 192.168.0.1 (192.168.0.1) from 192.168.0.4 : 4000(4028) bytes
of data.
Warning : time of day goes back, taking countermeasures.
4008 bytes from 192.168.0.1 : icmp_seq=1 ttl=128 time=7.218 msec
4008 bytes from 192.168.0.1 : icmp_seq=2 ttl=128 time=7.194 msec
4008 bytes from 192.168.0.1 : icmp_seq=3 ttl=128 time=8.381 msec
4008 bytes from 192.168.0.1 : icmp_seq=4 ttl=128 time=7.161 msec
4008 bytes from 192.168.0.1 : icmp_seq=5 ttl=128 time=7.175 msec
~
--- 192.168.0.1 ping statistics ---
23 packets transmitted, 22 packets received, 4% packet loss
round-trip min/avg/max/mdev = 7.116/8.363/19.669/2.569 ms
[root@kugstone /root]#
```

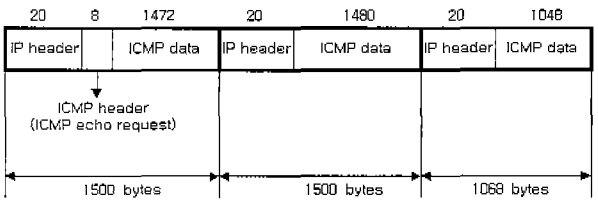
(그림 2) IP Fragmentation을 위한 ICMP 데이터 전송

이때 tcpdump 유틸리티를 이용하여 패킷을 모니터링 하면 (그림 3)과 같다. tcpdump의 막강한 packet filter 기능은 현재 로컬 네트워크 상에서 전송 중인 있는 특정한 패킷들을 실시간으로 기록해 줄 수 있으며, 이를 이용하여 네트워크에서 벌어지는 일들을 네트워크 관리자가 원하는 대로 선택하여 볼 수 있게 해 준다. 또한, 시스템 관리자들에게는 로컬 사용자의 외부로의 커넥션(Connection)들을 감시하고, 또한 특정 침입자가 침투 경로로 자주 이용하는 호스트, 혹은 원하지 않는 호스트로부터의 커넥션을 실시간으로 감시할 수 있는 기능을 제공한다[13].

```
[root@kugstone /root]# tcpdump
Kernel filter, protocol ALL, TURBO mode (575 frames),
datagram packet socket
tcpdump : listening on all devices
23 : 28 : 37.795126 eth0 > 192.168.0.4 > 192.168.0.1 : (frag 22873 :
1048@2960) (DF)
23 : 28 : 37.795126 eth0 > 192.168.0.4 > 192.168.0.1 : (frag 22873 :
1480@1480+) (DF)
23 : 28 : 37.795126 eth0 > 192.168.0.4 > 192.168.0.1 : icmp : echo
request
(frag 22873 : 1480@0+) (DF)
~
87 packets received by filter
[root@kugstone /root]#
```

(그림 3) tcpdump 유틸리티를 이용한 패킷 필터링

Ethernet 네트워크를 통하여 전송되기 전의 데이터 그램은 (그림 4)와 같이 20바이트의 IP헤더와 8바이트의 ICMP 헤더, 그리고 4000바이트의 ICMP 데이터를 가진 총 4028바이트의 데이터 그램으로 구분된다. 하지만 Ethernet를 통하여 전송되기 위해서는 Ethernet의 MTU 즉, 1500바이트를 넘을 수 없으므로 1500바이트 또는 그보다 작은 fragment로 분할하여 전송되게 된다[10].



(그림 4) ICMP 패킷 구조

(그림 3)에서 tcpdump에 의해 모니터링 된 첫 fragment의 내용에서 22873은 fragment ID, 1480은 Data Length, 0는 Fragment Offset, +는 MF flag가 1로 setting되어 있음을 보여준다. 여기에서 한 가지 중요한 사실은 tcpdump에서 첫 번째 패킷에서 나왔던 "ICMP : echo request" 패킷이 두 번째 fragment부터는 보이지 않는 다는 것이다. TCP나 UDP 패킷인 경우 목적지 포트번호 정보도 가지지 않는다. 이처럼 오직 첫 번째 fragment에만 TCP, UDP, 또는 ICMP 헤더가 포함되어 있어 패킷 필터링 장치에서 첫 번째 fragment만 차단되는 경우가 많기 때문에 fragment ID를 이용하여 각 session의 상태를 유지해야 한다는 것을 알 수 있다.

3. IP Fragmentation을 이용한 공격유형

IP Fragmentation은 큰 패킷을 전송하기 위해 발생되는 정상적인 과정이지만 공격자는 이 fragment를 조작하여 패

킷필터링 장비나 침입차단시스템을 우회하거나 서비스거부 공격을 유발시킬 수 있는데 이러한 공격의 대표적 유형은 다음과 같다.

3.1 IP Spoofing

TCP/IP 프로그램은 매우 유동적이며 사용하기 편리하지만 보안 측면에서는 아래와 같은 보안 문제점을 가지고 있다[14].

- 호스트의 인증 문제로 IP는 호스트의 인증을 IP의 출발지 주소만으로 수행하기 때문에 출발지 호스트의 IP 주소만을 속일 수 있다면 다른 호스트에 연결을 맺을 수 있다. Berkeley Unix의 rlogin, rsh, rexec 들이 그 예이다.
- 순서 번호(Sequence Number)의 생성 문제로 TCP Spec에서는 순서 번호가 초당 250,000번, 즉 4 msec마다 한번씩 증가시키도록 하고 있다. 그러나 Berkeley에서 구현된 TCP의 순서 번호는 4.2 BSD에서는 초당 128만큼 증가하고 4.3 BSD의 경우에는 초당 128,000만큼씩 증가한다. 즉, 초당 한번씩밖에 증가하지 않는다.

위와 같은 약점의 하나를 공격하는 IP Spoofing은 1985년 Morris에 의해 아이디어가 처음 제시되었고, 실제로 1995년도 San Diego Supercomputer Center를 해킹 하는데 Kevin Mitnick이 사용하기도 하였는데, 이 사건 이후로 이 해킹기술을 IP Spoofing라는 용어로 불리게 되었으며 현재까지 TCP/IP 약점을 이용한 공격 유형은 다음과 같다.

- 순서제어번호 추측(Sequence Number Guessing)
- 반(Half)접속시도 공격(SYN flooding)
- 접속 가로채기(Connection Hijacking)
- RST를 이용한 접속 끊기(Connection Killing by RST)
- FIN을 이용한 접속 끊기(Connection Killing by FIN)
- SYN/RST 패킷 생성공격(SYN/RST Generation)
- 네트워크 데몬 정지(Killing the Inetd)
- TCP 윈도우 위장(TCP Window spoofing)

그러나 일반적으로 IP Spoofing이란 Kevin Mitnick이 사용한 방법을 의미하며 순서제어 번호추측 공격, 반(Half)접속시도 공격 등이 함께 사용되는 고난도의 수법으로 볼 수 있다. 외부에서 들어오는 패킷 중에서 출발지 IP주소(Source IP Address)에 내부망 IP주소를 가지고 있는 패킷을 라우터 등에서 패킷 필터링을 사용하여 막아낼 수 있다. 그러나 내부 사용자에 의한 공격은 막을 수 없으므로 각 시스템에서 TCPwrapper, SSH(Secure Shell) 등을 설치해서 운영하고, rsh, rlogin 등과 같이 패스워드의 인증 과정이 없는 서비스를 사용하지 않는 것이 바람직하다.

3.2 Ping of Death

TCP/IP 프로토콜에서 IP 패킷의 최대 길이는 6만5,535까지로 제한되어 있다. 따라서 실제로 많은 시스템의 IP 패킷을 처리하는 코드는 이 같은 최대 길이를 가정하여 구현되어 있으며, 대부분의 시스템이 규정 길이보다 큰 패킷을 전송할 수 없도록 설정되어 있다. 그러나 윈도우 시스템을 포함한 일부 시스템에서는 이 같은 제한이 없어 규정된 길이 이상의 IP 패킷을 전송할 수 있다. 이 경우 최대 길이를 가정하여 구현된 시스템들은 IP 패킷 처리 코드의 버퍼가 초과되어 결과적으로 표준에 규정된 길이 이상으로 큰 IP 패킷을 전송함으로써 이 패킷을 수신 받은 OS에서 이 비정상적인 패킷을 처리하지 못함으로써 서비스거부공격(Denial of Service Attack)을 유발하도록 하는 방법이다.

통상적으로 이러한 공격 방법은 가장 손쉽게 IP 패킷으로 전송할 수 있는 ping 프로그램을 이용하여 수행되는데, ping 프로그램은 실제 ICMP ECHO Request 패킷을 상대방에게 전송한다. IP 패킷의 헤더는 기본적으로 특별한 옵션을 사용하지 않았을 경우에 56에서 64 바이트가 사용되며, ICMP ECHO Request 패킷은 8바이트의 ICMP 헤더를 사용하므로 실제 데이터 길이의 최대값은 65535 - 20 - 8 = 65507바이트가 된다. 따라서 ping 패킷의 최대 길이를 제한하지 않는 시스템에서는 (그림 5)와 같은 간단한 명령으로 공격을 수행할 수 있다[10].

```
[root@kugstone /root]# ping -l 65510 192.168.0.1
```

(그림 5) ping 프로그램에 의한 서비스 거부 공격

3.3 ICMP 공격

ICMP은 인터넷프로토콜에서 문제가 생기면 보고해주는 프로토콜로서, 예를 들어 네트워크 접속문제인 "Echo Reply", "Destination Unreachable" 혹은 네트워크 라우팅 문제인 "Redirect" 등을 보고한다. 여기에서 잘 접속되어 있는 시스템이 문제가 있는 양 ICMP 메시지를 만들어 접속을 거부할 수 있으며, ICMP Redirect를 의도적으로 만들어 네트워크 라우팅 자체를 혼란에 빠지게 할 수 있다. 외부에서의 ICMP 공격 시도를 방화벽(Firewall) 개념에서 감시하거나 막을 수 있다.

3.4 Packet Sniffering

최근에 널리 쓰이고 있는 방법으로, tcpdump, snoop, sniff 등과 같은 네트워크 모니터링 도구를 이용해 네트워크 내에 전송중인 패킷의 내용을 분석해 정보를 알아내는 것

이다. Ethernet은 로컬 네트워크내의 모든 호스트가 같은 선(wire)을 공유하도록 되어 있기 때문에 같은 네트워크내의 컴퓨터는 다른 컴퓨터가 통신하는 모든 트래픽을 볼 수 있다. 하지만 Ethernet을 지나는 모든 트래픽을 받아들이면 관계없는 트래픽까지 처리해야 하므로 효율적이지 못하고 네트워크의 성능도 저하될 수 있다. 그래서 Ethernet Interface(LAN 카드)는 자신의 MAC address를 갖지 않는 트래픽을 무시하는 필터링 기능을 가지고 있으며, 이 필터링 기능은 자신의 MAC address를 가진 트래픽 만을 보도록 하는데, 네트워크에 연동돼 있는 호스트 뿐 만 아니라 외부에서 내부 네트워크로 접속하는 모든 호스트가 그 대상이 된다. 또한 Ethernet 인터페이스에서 모든 트래픽을 볼 수 있도록 하는 기능을 설정할 수도 있는데 이를 "promiscuous mode"라 하는데, 스니퍼는 Ethernet Interface를 이러한 "promiscuous mode"로 설정하여 로컬 네트워크를 지나는 모든 트래픽을 도청할 수 있게 된다[15].

이외에도 최초의 fragment를 아주 작게 만들어서 네트워크의 침입 탐지 시스템이나 패킷 필터링 장비를 우회해서 공격하는 Tiny Fragment 공격, fragment들이 재 조합될 때의 허점을 이용한 Fragment Overlap 공격, IP를 가로채는 IP Hijacking등이 있다[10].

4. 실시간 접근로그 구현 및 평가

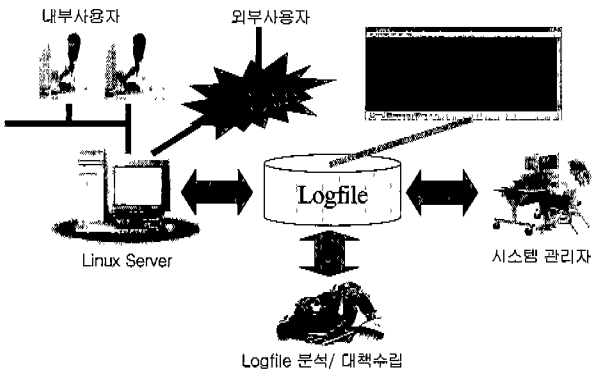
네트워크 공격을 하기 위한 첫 번째 행동은 정보수집을 위한 방법으로 표준 프로토콜 방식인 TCP/IP 기반의 ping, finger, host와 같은 명령어들이 사용된다. 본 장에서는 이때 사용하는 ICMP프로토콜을 이용하여 본 논문에서 제안하는 실시간 접근 로그 시스템의 설계 및 구현에 대하여 기술한다. 본 시스템 구현 환경으로는 영문 레드헷 리눅스 7.1 환경에서 ansi-C로 작성하였으며, 컴파일러는 gcc를 사용하였다. 프로그램 실행은 별도의 백그라운드(Background) 작업 없이 실행과 동시에 데몬(Demon)으로 작동하도록 하였다.

4.1 실시간 접근 로그 프로그램 구조

본 논문에서 제안한 실시간 접근 로그 프로그램(Real Time Access Log Program, 이하 RTAL)은 (그림 6)에서 볼 수 있듯이 세 부분으로 구성된다. 네트워크 상에 존재하는 자신의 컴퓨터와 관련이 있는 모든 패킷의 정보를 필터링하는 부분, 이를 바탕으로 실시간 접근 로그를 생성하는 부분과 해당 패킷 정보를 종합적으로 분석 할 수 있도록 구성되어 있다.

4.2 ICMP 분석 및 로그 생성 모듈

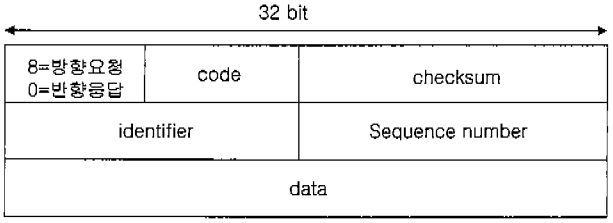
ICMP는 호스트 서버와 인터넷 게이트웨이 사이에서 메시지를 제어하고 에러를 알려주는 프로토콜로서 RFC 792에 정의되어있다. ICMP는 IP 데이터 그램을 사용하지만, 메시지는 TCP/IP 소프트웨어에 의해 처리되며, 응용프로그램 사용자에게 직접 분명하게 보이지는 않는다. 일례로서, ping 명령어는 인터넷 접속을 테스트하기 위해 ICMP를 사용한다. ICMP는 5개의 오류 메시지와 4개의 정보 메시지를 정의하며, 가장 일반적으로 이용되는 ICMP 질의 메시지는 ping 프로토콜을 구현하는데 사용되는 ICMP 메시지들로 구성된다.



(그림 6) RTAL 프로그램 구조

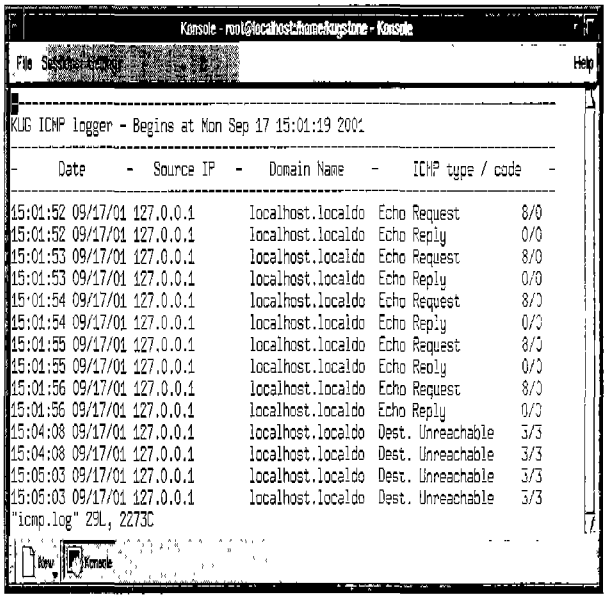
```
FILE *fkug ;                    /* Simple log file pointer */
FILE *fpraw ;                   /* Raw log file poinetr */
char buf[ 65535] ;              /* The buffer we use to read a
                                   packet */
char *type ;                    /* ICMP type string */
struct kug_iphdr *iph ;         /* IP header pointer */
struct kug_icmphdr *ich ;       /* ICMP header pointer */
struct in_addr addrs ;
struct in_addr *mask ;          /* Mask to apply for IP addresses to
                                   ignore */
struct in_addr *ignoremask ; /* Mask to apply for ignored
                                   packets */
struct tm *ltm ;                /* Used by localtime(3) */
time_t t ;                      /* Used by time(2) */
char *ti ;                      /* Used by ctime(3) */
int ch ;                        /* For command line parsing */
int iphlen ;                    /* IP header length (Used to skip */
```

(그림 7) ICMP 패킷 분석 및 로그 파일 생성을 위한 변수 및 구조체 선언

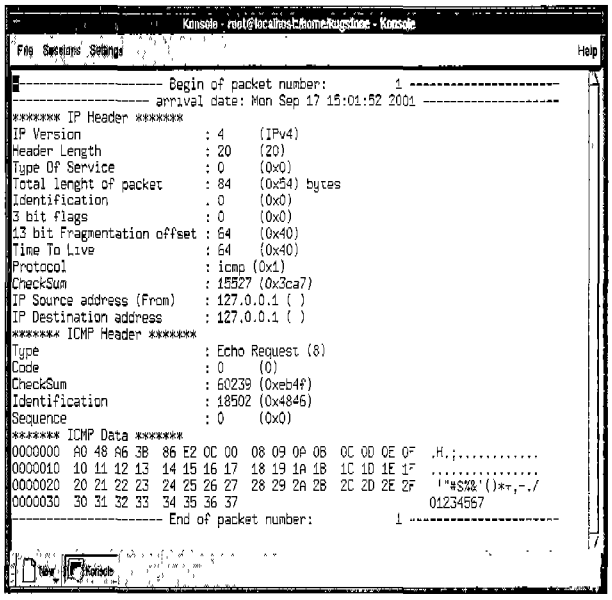


(그림 8) ICMP Echo Request 및 Reply 형식

호스트나 라우터가 ICMP Echo Request메시지를 수신한다면 ICMP Echo Reply로 응답한다[1]. 이러한 메시지에 포함된 패킷을 분석하여 로그 파일을 생성하기 위해서 (그림 7)과 같은 변수와 구조체 함수를 사용한다. (그림 8)은 ICMP Echo Requested 및 Replay 형식을 보여주고 있으며, RTAL 프로그램을 이용하여 ICMP 패킷을 필터링 한 결과를 (그림 9)와 같이 보여준다. (그림 10)는 자신의 시스템과 관련이 있는 각각의 패킷정보와 포함하고 있는 데이터를 분석하여 (그림 8)의 ICMP 형식에 맞추어 실시간 접근 로그를 생성한다.



(그림 9) ICMP 패킷 필터링 Log 파일



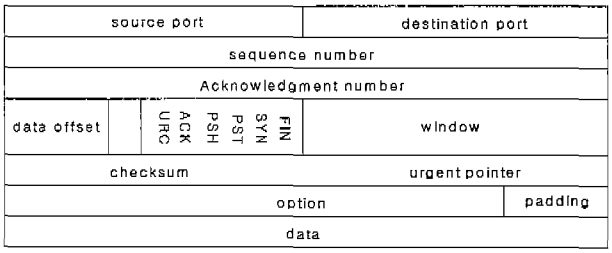
(그림 10) ICMP 패킷 분석을 통한 실시간 접근 로그 파일 생성 결과

4.3 TCP 분석 및 로그 생성 모듈

많은 응용 서비스들, 즉 FTP, TELNET, SMTP, X.400 등과 같은 기능을 구현하기 위해 TCP를 사용한다. TCP는 전송되는 데이터를 연속된 Octet Stream으로 보는 Stream 중심의 데이터 전달 서비스를 제공한다. 한 TCP 사용자로부터 다른 사용자로 전송되는 Octet들은 보내진 순서대로 목적지 호스트에 나타난다. TCP에서는 동일한 데이터 Stream이 목적지 호스트에 모두 나타나거나, 아니면 연결이 해제되어 양쪽 TCP 사용자에게 오류 사실이 통보된다. 이러한 메시지에 포함된 패킷을 분석하여 로그 파일을 생성하기 위해서 (그림 11)과 같은 변수와 구조체 함수를 사용한다.

```
FILE *fkug ;                    /* Simple log file pointer */
FILE *fpraw ;                   /* Raw log file poinetr */
char buf[ 65535] ;              /* The buffer we use to read a
                                   packet */
struct kug_iphdr *iph ;         /* IP header pointer */
struct kug_tcphdr *tcph ;       /* TCP header pointer */
struct in_addr addrs ;
struct in_addr *mask ;          /* Mask to apply for IP addresses
                                   to ignore */
struct in_addr *ignoremask ;    /* Mask to apply for ignored packets */
struct tm *ltm ;                /* Used by localtime(3) */
time_t t ;                      /* Used by time(2) */
char *ti ;                      /* Used by ctime(3) */
int flag_mask ;                 /* Only log packets that matches
                                   tcpflagmask */
PERRO_U8 tcpflagmask ;          /* The TCP flag mask (to ignore
                                   packets) */
int ch ;                        /* For command line parsing */
int iphlen ;                    /* IP header length (Used to skip */
```

(그림 11) TCP 패킷 분석 및 로그 파일 생성을 위한 변수 및 구조체 선언

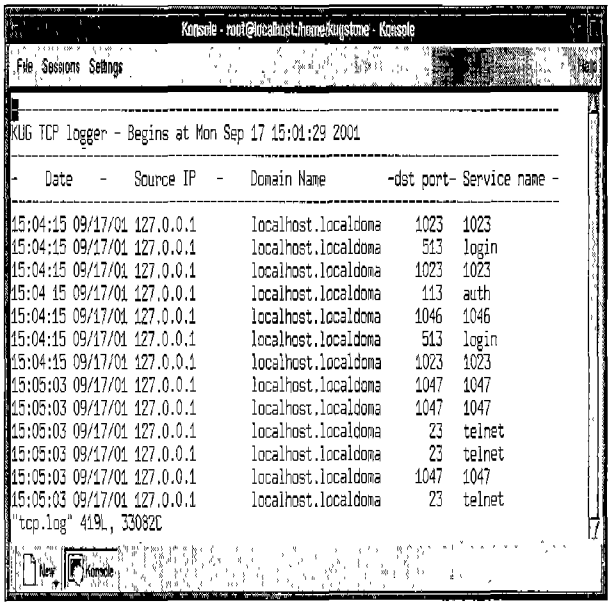


(그림 12) TCP 세그먼트 형식

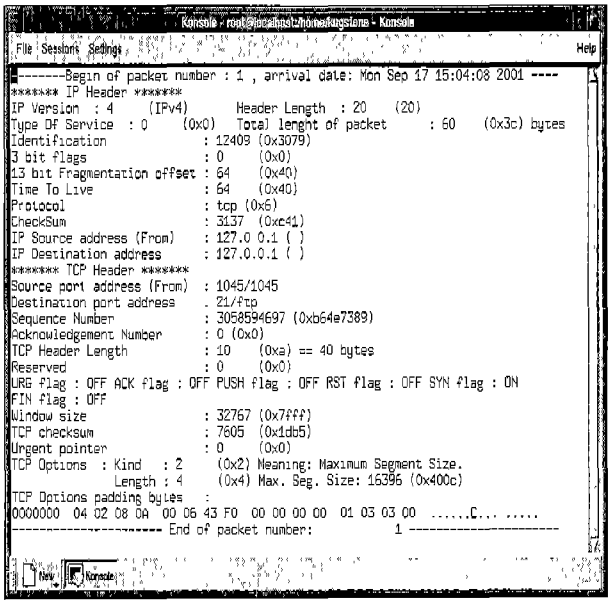
(그림 12)은 이러한 TCP 형식을 보여주고 있으며, (그림 13)는 TCP와 관련된 패킷을 필터링 한 결과를 보여주며 (그림 14)는 각각의 패킷을 분석하여 RTAL 프로그램이 생성한 실시간 접근 로그 파일을 보여준다.

5. 결 론

IP Fragmentation은 이 기종의 네트워크 환경에서 IP 패



(그림 13) ICMP 패킷 필터링 Log 파일



(그림 14) TCP 패킷 분석을 통한 실시간 접근 로그 파일 생성

킷의 효율적인 전송을 보장해주고 있지만 앞서 살펴본 것과 같이 몇 가지의 보안문제를 가지고 있으며, 많은 패킷 필터링 장비나 침입탐지시스템, 그리고 각 운영체제의 IP 스택이 IP Fragmentation 재 조합을 적절히 처리하지 못하고 있다.

본 논문에서는 불법으로 특정 시스템을 침입하기 위해서는 먼저 상대호스트 정보를 알아내어야 하는데 이때 사용하는 방법이 바로 표준 프로토콜 방식인 TCP/IP 기반의 ping, finger, host 과 같은 명령어들이다. 이와 같은 명령어를 이용할 때 외부에서 접근한 사용자에 대해서 리눅스 시스템은 로그를 남겨놓지만, 외부에서 사용한 명령어들은 로그를 남겨놓지 않아 이러한 보안 문제점을 보완하기 위해 ICMP, TCP에 관련된 패킷을 분석하여 실시간 접근 로그를 생성하여 침입 탐지 및 대책을 강구할 수 있는 프로그램을 구현 및 평가하여 타당성을 검증하였다. 그 결과 다양한 형태로 조개진 fragment 들을 실시간으로 분석하여 시스템 관리자가 의사결정을 할 수 있는 것과 동시에 IP Fragmentation에 관련된 다양한 공격유형에 대처할 수 있도록 패킷 정보를 분석해 내는 기능을 제공하였다.

향후의 연구할 방향으로는 본 논문에서 제안된 모델에 근거하여 IP Fragmentation에 관련된 공격 유형을 재정립하고, 각 공격유형에 대처할 수 있는 데이터베이스를 구축하여 시스템 스스로 대처할 수 있는 시스템을 개발하여 리눅스 보안을 더한층 강화하는 것이다.

참 고 문 헌

[1] James Martin, joe Leben, "Tcp/Ip Networking : Architecture, Adminstration, and Programming," Prentice Hall, August, 1994.

[2] Chris Hare, Karanjit Siyan, "Internet Firewalls and Network Security," 2nd Bk&Cd edition, New Riders Publishing, August, 1996.

[3] N. Derek Arnold, "Unix Security A Practical Tutorial," McGraw-Hill, Oct. 1995.

[4] Graham Class, "Unix for Programmers and Users A Complete Guide," McGraw-Hill, Aug. 1994.

[5] Stephen Northcutt, "Network Intrusion Detection An Analyst's Handbook," New Riders Publishing, 2000.

[6] 이상훈, 국경완, "유닉스 시스템 이론과 응용", 사이텍미디어, Jul. 2001.

[7] 이상훈, 국경완, "실시간 파일시스템 접근로그 감시를 통한 리눅스 보안강화에 관한 연구", 11 쪽, 한국전자통신연구원, Jul. 2001.

[8] http : //www.gyro.pe.kr/lecture/internet/17.htm.

[9] http : //ise.yonsei.ac.kr/yhlee/kvalley/152/3_6.html.

[10] 정현철, "IP Fragmentation을 이용한 공격기술들", 한국 정보보호 센터, 2001.

[11] 이현우, "네트워크 공격기법의 패러다임 변화와 대응방안", 한국 정보보호 센터, 2000.

[12] 임채호, "중요정보통신망 해킹시 침입자기법 분석과 대응", 한국 정보보호 센터, Jan. 1999.

[13] http : //wonie.co.kr/security/tcpdump.html.

[14] IP spoofing 공격과 대책, 한국 정보보호 센터, 1996.

[15] 박현미, 신은경, 이현후, "네트워크 스니핑 기술 및 방지대책", 한국 정보보호 센터, 2000.

국 경 완

e-mail : kugstone@hitel.net
1992년 금오공과대학 전자계산학(공학사)
1993년~1999년 전산실장(UNIX 시스템 관리자)
2001년 국방대학교 전산정보(공학석사)
2002년~현재 육군정보학교 전산실장
관심분야 : 리눅스/자바 프로그래밍, 객체지향, 소프트웨어 공학

이 상 훈

e-mail : hoony@kndu.ac.kr
1978년 성균관 대학교 전자공학과(공학사)
1989년 연세대학교 전자계산학(공학석사)
1997년 일본 Kyoto대학 정보공학(공학박사)
1998년 충남산업대학교 멀티미디어과 교수
2000년~현재 국방대학교 전산정보학과 교수
관심분야 : 협조작업처리(CSCW), 멀티미디어 데이터베이스, 멀티미디어, 객체지향 데이터베이스