Journal of Korea Multimedia Society Vol. 9, No. 8, August 2006(pp. 1037-1044)

SSFNet을 이용한 대규모 네트워크상에서의 보안 시뮬레이션을 위한 방화벽과 IPS모듈의 모델링 및 구현

김용탁[†], 김태석[††], 권오준[†††]

요 약

실제 대규모 네트워크에서 사이버 공격으로 인한 보안시스템의 성능을 검증하기는 어렵다. 일반적으로 새로운 보안시스템이나 공격기법은 시뮬레이션을 통해 검증을 한다. 본 논문에서는 보안시스템 및 공격기법을 시뮬레이션 하기 위해 SSFNet을 사용하였다. SSFNet은 프로세스 기반 사건 중심 시뮬레이션 시스템이면서, 대규모 네트워크를 쉽게 표현할 수 있는 장점을 가지고 있다. 하지만 보안시스템이나 패킷을 조작할 수 있는 API를 제공하고 있지 않다. 본 논문에서는 보안 시스템으로 주로 사용하고 있는 방화벽과 IPS 클래스를 개발하여 SSFNet 구성요소로 추가하였다. 방화벽은 패킷 필터링 기반 보안 방화벽 시스템을 모델링하였다. IPS는 탐지 기능과 이상 패킷에 대한 드롭 기능을 가지고 있다. 확장된 SSFNet내에 네트워크를 모델링하여 방화벽과 IPS의 기능을 검증하였다. 방화벽은 패킷의 주소와 포트의 규칙을 통해 패킷을 차단하였다. 그리고 라우터에 기술된 IPS의 로그화면을 통해 패킷의 상태와 이상 패킷이 차단되는 것을 확인하였다

Modeling and Implementation of Firewall and IPS for Security Simulation on Large-scale Network Using SSFNet

Yong-Tak Kim[†], Tai-Suk Kim[††], Oh-Jun Kwon[†††]

ABSTRACT

It's difficult to check cyber attacks and the performance of a security system in a real large-scale network. Generally, a new security system or the effect of a new security attack are checked by simulation. We use SSFNet to simulate our security system and cyber attack. SSFNet is an event-driven simulation tools based on process, which has a strength to be capable of expressing a large-scale network. But it doesn't offer any API's which can manipulate not only the related function of security but also the packet. In this paper, we developed a firewall and IPS class, used for a security system, and added to them components of SSFNet. The firewall is modelled a security system based on packet filtering. We checked the function of the firewall and the IPS with network modelled as using our SSFNet. The firewall blocks packets through rules of an address and port of packets. The result of this simulation shows that we can check a status of packets through a log screen of IPS installed in a router and confirm abnormal packet to be dropped.

Key words: SSF(Scalable Simulation Framework)Net, IDS : Intrusion Detection System(침입탐지시스템), IPS : Intrusion Prevention System(침입방지시스템)

※ 교신저자(Corresponding Author) : 김태석, 주소 : 부산시 부산진구 엄광로995(614-714), 전화 : 051)890-1707, FAX : 051)890-1724, E-mail : tskim@deu.ac.kr
접수일 : 2006년 3월 7일, 완료일 : 2006년 5월 30일
[†] 준회원, 동의대학교 컴퓨터응용공학과
(E-mail : 63164@deu.ac.kr)
[††] 종신회원, 동의대학교 컴퓨터소프트웨어공학과
[†††] 종신회원, 동의대학교 컴퓨터소프트웨어공학과
(E-mail : ojkwon@deu.ac.kr)

1. 서 론

컴퓨터 기술의 발달과 더불어 인터넷의 발전은 데이터 전송 속도의 가속화 및 대용량의 데이터 전송 등의 기술을 증가시켜 업무 효율을 향상시킴으로서, 현대인 생활의 질을 높여 주는 긍정적인 효과를 가져왔다. 하지만 인터넷 확장으로 인하여 외부인의 시스템 불법 침입, 불법적인 사용, 컴퓨터바이러스 및 서비스 거부 공격 등의 역기능들이 계속해서 증가되어 그 피해가 심각한 수준이다.[1,2]

네트워크상에서의 침입 시도는 해가 갈수록 증가되고, 기술은 다변화되고 있으며, 악의적인 사용자들에 의한 독창적이고 새로운 침입 방식의 개발은 전체 네트워크의 성능을 저하시키는 문제점을 가져왔다. 이러한 공격에 대한 대응책이 절실하게 요구되면서, 사이버 테러로부터 시스템을 보호하기 위해 침입차단시스템, 침입탐지시스템, 침입방지시스템, 가상사설망 등 여러 보안 제품들이 부각되었다.

하지만 현 보안시스템인 방화벽은 단순한 차단 기능, 알려진 공격패턴 감시 등을 통해 공격을 감지하지만 Nimda나 CodeRed 같은 새로운 공격을 막기에는 역부족이다. IDS 또한 알려지지 않은 공격에 대한 탐지가 곤란하고 내부 공격자를 막기에도 어려움이 있다. 특히, 침입탐지의 오판에 따른 시간, 인적, 재정 낭비도 문제점으로 지적된다. 이에 반해 침입방지시스템인 IPS는 알려지지 않은 공격에 대해서 적절하게 대응하며, 명백한 공격은 사전 방어를 취한다. 또 웜과 바이러스 등의 침입을 네트워크 계층에서 차단시킴으로 보안 인프라와 네트워크 영향을 제거한다.[3]

이처럼 IPS는 Worm 바이러스와 해킹으로부터 유발되는 네트워크 서비스 장애로부터 벗어날 수 있고 부가적으로 유해한 트래픽을 사전 차단함으로서 인터넷 및 네트워크 자원을 효율적으로 사용할 수 있다.

본 논문에서는 SSFNet[4-6]의 IP 클래스를 확장하여 방화벽·침입탐지시스템 기반의 IPS를 추가하였다. Snort 기반의 탐지 기능을 가진 IDS[7-10]와 차단 기능을 가진 방화벽을 결합한 IPS를 모델링하고 구현하였다. 그리고 SSFNet에서 지원하지 않는 패킷 조작기(PM: Packet Manipulator)를 사이버 공격 프로그램을 만들기 위해 추가하였다.

확장한 SSFNet을 사용하여 IPS모듈을 올린 라우터를 통과하는 트래픽을 실시간으로 모니터링 할 수 있으며, 규칙 집합과 비교하여 이상 패킷을 차단할 수 있다. 규칙 집합은 패킷이 기본적인 내용으로 프로토콜 타입, 출발지와 목적지 IP주소, 포트 등으로 구성되며, 규칙 헤더의 내용을 비교 판단할 수 있는 규칙 옵션으로 이루어진다.[11-12]

확장한 SSFNet을 사용하여 시뮬레이션을 위한 네트워크 환경을 모델링하고, 추가한 보안 모듈인 IPS를 각 라우터에 설치하여 기능을 검증하였다. 확장된 SSFNet을 사용하여 대규모 네트워크 환경의 보안 시스템을 효율적으로 설치하고 관리할 수 있는 장점을 가지고 있다.

본 논문의 구성은 다음과 같다. 2장에서 관련 논문으로 침입탐지시스템에 대해 기술한다. 3장에서는 침입차단시스템인 방화벽의 모델링 구현하고 기능을 검증하였다. 4장에서는 IPS를 모델링하여 설계하였으며, 5장에서는 패킷을 조작하여 구현한 IPS의 기능을 검증하였다. 끝으로 6장에서 결론을 기술하였다

2. 관련 연구

2.1 IDS 구조 및 동작절차

보안시스템 중 하나인 IDS를 소프트웨어로 구현하기 위해 Snort엔진을 기본 모델로 많이 사용하고 있다. 본 논문에서도 Snort 기본 메커니즘을 사용하여 IDS를 모델링하였다. 그림 1은 Snort의 구조를 보여주고 있다.

네트워크의 패킷은 Snort의 스니퍼를 사용하여 네트워크를 도청할 수 있다. 네트워크 스니퍼를 사용하면 어플리케이션 또는 하드웨어 장치에서 네트워크의 트래픽을 볼 수 있다. 인터넷의 대부분은 IP 트래픽이라 할 수 있다. 전처리기는 원본 패킷을 받아들여서 특정한 플러그-인으로 그 패킷을 보낸다. 이들 플러그-인은 패킷에서 특정한 종류의 행위를 찾는

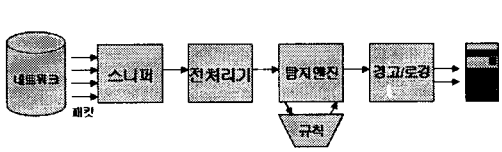


그림 1. Snort 구조

다. 패킷의 특정한 종류의 '행위'를 찾으면 그 패킷은 탐지 엔진으로 전송된다. 탐지 엔진은 전처리기와 플러그-인으로부터 오는 데이터를 받아서 여러 규칙과 비교한다. 만약 패킷과 일치하는 규칙이 있다면 그 패킷은 경고 처리기로 전달한다.

2.2 IDS 규칙형식

패킷에 대한 처리 방법에 대해서는 각 패킷에 대한 행동을 결정하는 데이터가 있어야 한다. Snort에서는 데이터베이스를 사용하여 특정 패킷에 대하여 규칙을 정의하였다. 표 1과 같이 규칙헤더와 규칙옵션으로 구성된다.

규칙헤더는 기본적으로 취할 행동(로그 또는 경고), 네트워크 패킷의 종류(TCP, UDP, ICMP 등), 출발지와 목적지 IP주소, 포트 등으로 이루어진다.

규칙 옵션은 패킷이 규칙과 일치하기 위해 포함해야 하는 내용이다. IDS는 기본적으로 규칙 문법을 가지고 있으며, 규칙은 종류별로 묶여 정기적으로 갱신이 된다.

표 1. 규칙 헤더와 옵션 구조

RuleHeader		
Name	Description	
no	규칙 번호	
proto_no	프로토콜 번호	
name	프로토콜 명시	
action	이 패킷에 대한 행동 결정	
srcAddr	송신자 IP주소	
RuleHeader		
srcPort	송신자 포트번호	
destAddr	목적지 IP 주소	
destPort	목적지 포트번호	
flag	규칙 옵션 수행여부 결정	
Rule Option		
Type	name	Description
TCP	tcp_falgs	SYN, FIN, ACK
	tcp_seq	시퀀스 번호
	tcp_ack	ACK의 번호
ICMP	icmp_type	메시지 형태
	icmp_code	메시지 코드
	icmp_id	메시지 ID
	icmp_seq	메시지 번호

많은 침입탐지시스템들은 Snort의 구조를 기반으로 하여 규칙헤더와 옵션구조를 테이블로 작성하여 패킷의 구조를 비교 분석한다.

3. SSFNet내의 방화벽 모듈 구현

3.1 모델링

침입차단시스템(방화벽 혹은 firewall)은 외부로부터의 불법적인 접근이나 해커의 공격으로부터 내부 네트워크를 방어하기 위해 내부 네트워크와 외부 네트워크 사이의 통로를 설치하여 두 네트워크간의 트래픽을 제어하기 위한 목적으로 구성된 가장 널리 이용되는 보안 시스템이다.

본 논문에서는 사이버 공격 때 네트워크의 행동변화를 시뮬레이션 하기 위해 SSFNet을 확장하여 방화벽을 모델링 할 수 있도록 구성요소를 추가하였다.[13] 방화벽은 동작하는 프로토콜 계층에 따라 분류할 수 있다. 본 논문에서는 모델의 3계층인 네트워크 계층과 4계층인 전송 계층에서 패킷 필터링 기능을 수행하는 방화벽시스템을 모델링하였다. 패킷 필터링 방화벽은 기본적인 단위인 패킷의 헤더 정보를 분석하여 미리 정한 규칙의 집합에 따라서 필터링을 한다. 패킷 필터 규칙은 필터링 형태(incoming, outgoing, forwarding 패킷 여부), 근원지 및 목적지의 IP주소와 포트번호, 프로토콜, 그리고 규칙과 일치하는 패킷의 처리방법(패킷 폐기 : 차단(drop), 패킷 통과 : 허용(accept))등으로 정의 된다.

방화벽의 기능을 모델링하기 위해 그림 2와 같이 SSFNet의 PrtocolSession에 IP ProtocolSession을 확장하여 SIP ProtocolSession 부분을 추가하였다.

IP ProtocolSession 클래스는 보안적인 내용을 가지고 있지 않다. 그래서 기존 클래스의 내용을 상속받은 보안 클래스의 개념을 가진 SIP 클래스를 작성

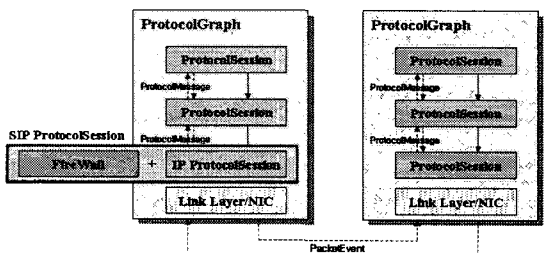


그림 2. 방화벽을 위한 SIP 프로토콜

하였다. SIP 클래스는 필터링 규칙을 가지고 있으며, 이 필터링 규칙에 준하여 패킷의 헤더를 분석하여 컴퓨터로 들어오는지, 나가는지 또는 전달되는지를 결정한다. 필터링 규칙과 일치하는 경우에는 허용하고, 그렇지 않는 경우에는 차단하도록 하였다.

방화벽을 시뮬레이션 할 경우에는 DML(Domain Modeling Lanuage)내의 라우터를 기술하는 부분에서 ProtocolSession부분의 IP 클래스를 확장한 SIP 클래스로 지정해야 한다. 그리고 패킷의 필터링 규칙(송·수신지 주소, 포트, 인터페이스 번호)도 함께 정의하고 있다. 그림 2는 라우터의 DML 기술 내용에 규칙에 대한 내용을 기술하였다.

```
router[ id 2
interface [ id 0 _extends .dictionary.100BaseT]
interface [idrange [from 1 to 2] buffersize 16000]
graph[     # SSF.OS.IP 대신에
           # SSF.OS.SIP 클래스로 정의하였다.
    ProtocolSession [name ip use FIREwall
      Firewall [
            Rule[ id 0
                  srcAddr 0.0.0.6
                  destPort *
                  destAddr 0.0.0.10
                  destPort TCP
              ]
            Rule[ id 1
              srcAddr 0.0.0.6
              destPort *
              destAddr *
              destPort ICMP
            ]
            Rule[ id 2
              srcAddr 0.0.0.10
              destPort *
              destAddr *
              destPort TCP
            ]
        ] # end of firewall
    ] # end of ProtocolSession
    ProtocolSession [name icmp use SSF.OS.OSPF]
  ] # end of Graph
]
```

그림 3. 방화벽을 모델링하는 DML 표현

3.2 기능 검증

방화벽 모듈의 기능을 검증하기 위해 그림 4와 같은 네트워크를 구성하였다. 서버와 클라이언트, 라우터를 각각 2대씩 설정하였다. 메시지 프로토콜로 TCP 사용하도록 DML의 내용을 기술하였다

본 시뮬레이션에서는 R2에 방화벽 모듈을 설정하였으며, 이를 R2의 DML내의 라우터 기술부분에서 IP 클래스 대신 SIP 클래스를 지정하였다. 패킷 흐름의 방향은 **Clnt11** 은 **Serv22**와 **Serv21**로 보내도록 설정하였고, **Clnt12**는 **Serv21**에만 패킷을 보내도록 설정하였다. 패킷의 조작을 위해 개발한 패킷 생성기에서 모든 패킷의 프로토콜 타입을 TCP로 설정하여 통신이 되도록 설정하였다.

그림 5는 그림 4의 네트워크에서 방화벽으로 설정한 R2를 통해 가는 패킷을 추적한 화면이다. DML내

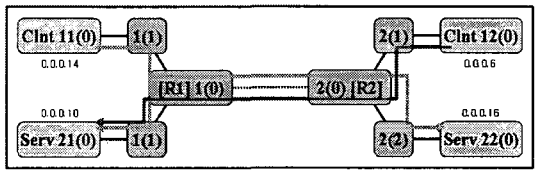


그림 4. 시뮬레이션을 위한 네트워크 구조

```
** Running for 1234000000000 clock ticks (== 1234.0 seconds sim time)
1.616458204 TCP host 11 src={0.0.0.14:10001} dest={0.0.0.10:1600}
Active Open
1.618720721 TCP host 21 src={0.0.0.10:1600} dest={0.0.0.14:10001} SYN
recvd
1.82214099 TCP host 12 src={0.0.0.6:10001} dest={0.0.0.10:1600} Active
Open
Drop!!
Drop!!
9.958811162 [ sid 1 start 1.616458204 ] client 11 srv 21(0) rcvd 10000000B
at 9589.62kbps - read() SUCCESS
9.958811162 TCP host 11 src={0.0.0.14:10001} dest={0.0.0.10:1600}
Active Close
9.961073679 TCP host 21 src={0.0.0.10:1600} dest={0.0.0.14:10001}
Active Close
9.961073679 TCP host 21 src={0.0.0.10:1600} dest={0.0.0.14:10001}
Passive Close
Drop!!
129.839234331 TCP host 11 src={0.0.0.14:10001} dest={0.0.0.10:1600}
2MSL timeout, connection closed
```

그림 5. 방화벽에 대한 시뮬레이션 결과

의 규칙(Source address의 주소가 0.0.0.10이고, TCP 프로토콜)에 따라, R2를 통과하는 host 21 0.0.0.10의 TCP 메시지가 목적지 주소 0.0.0.14로 도착하지 못하고, 차단된 것을 확인할 수 있다. 이와 같은 시뮬레이션을 통해 방화벽 기능 타당성 검증에 문제가 없음을 확인 할 수 있었다.

4. SSFNet의 IPS 모듈 구현

침입방지시스템(IPS)의 기능은 외부 네트워크에서 내부 네트워크로의 불법적인 침입을 방지하는 것이다. 기존 보안시스템인 방화벽은 단순 차단 기능과 이미 알려진 공격패턴 감시하는 기능만을 가지고 있으며, 침입탐지시스템의 경우 네트워크 트래픽의 이상 징후를 탐지하여 침입에 대한 경보메시지 및 감사기록을 통해 이후에 침입의 탐지에 대응할 수 있도록 개발되어져 있다.

본 논문에서는 대규모 기반의 네트워크를 모델링 할 수 있는 SSFNet 구조에 보안 모듈을 설계하고, 구현하였다. 보안 구조를 가진 확장된 SSFNet 시뮬레이터는 보안 관련 연구를 직접 시뮬레이션 할 수 있는 장점을 가지고 있다.

본 논문에서는 IPS의 기능을 분석하여 SSFNet 패킷지로 구현하였다. SSFNet 시뮬레이터에서 사용되는 IPS의 기능은 패킷 필터링과 실시간 트래픽을 로그화면을 통해 확인 할 수 있도록 하였으며, 규칙을 기반으로 하여 패킷의 이상 유무를 확인할 수 있다. 그리고 유해한 패킷의 경우, 규칙에 의해 내부 네트워크로 패킷이 들어오는 것을 차단할 수 있는 기능을 가지고 있다.

4.1 보안 구조 및 설계

본 논문에서 제안한 IPS의 모델은 네트워크 기반의 IDS의 탐지 기능과 방화벽의 패킷 필터링 기능을 결합한 모듈로 구현하였다. 그림 6의 IPS는 방화벽·IDS을 기반으로 동작하도록 모델링하였다. 그림 6은 제안한 IPS의 기본 구조이며, 동작 절차에 대한 내용이다. IPS의 참조 모델로 Snort를 사용하였다. 패킷 스니퍼에 보안 모듈을 가진 노드로 들어오는 모든 패킷을 캡처하고, 전 처리기에서 IP 클래스를 통과할 때 정의한 큐 속에 복사되도록 하였다. 큐 속에 복사된 패킷은 IPS 엔진에서 Rule Header와 Rule option

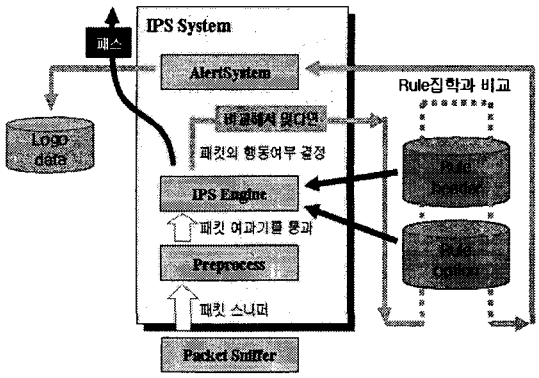


그림 6. IPS 동작 방식

의 내용과 비교되며, 패턴이 일치할 경우에는 패킷을 직접 차단할 수 있도록 구현하였다. 그리고 이상 징후의 패킷인 경우, 경보 메시지를 로그 데이터를 남겨 분석하여 대응할 수 있도록 하였다. 그리고 패킷의 경우에는 규칙 집합과 비교되고, 필터링 되어 내부 네트워크로 전달이 된다.

Rule Header와 Rule option의 구조는 표 1을 참조하였으며, SSFNet의 보안 모듈에 들어오는 패킷은 MySQL로 테이블로 작성된 규칙과 직접 비교되도록 설계하였다.

4.2 클래스 구조

SSFNet내의 IPS 클래스 구조는 기존 IP 클래스의 멤버 변수와 멤버 함수를 그대로 상속받아 사용하였다. 보안 내용을 추가한 상속된 IP클래스를 SIP 클래스로 명명하여 사용하고 있다. 그리고 본 논문에서는 SIP 클래스를 사용하여 독립된 구성요소로 이루어진 IPS 패키지로 SSF의 SSF.OS에 추가하여 구현하였다.

SSFNet은 SSF를 기반으로 지원되는 라이브러리이다. SSF의 구성은 OS측면과 Net측면으로 나누어진다.

SSF.OS Package는 네트워크 Protocol의 모델링과 시뮬레이션을 위한 구조를 가지고 있다. 이는 x-kernel의 구조를 기초로 설계되었지만 보다 간편하다. SSF.OS의 구조는 ProtocolSession, Protocol Message 그리고 Protocol Graph의 세 가지 클래스를 기반으로 구성되어 있다.

기본 클래스로서 Protocol Service 요청과 들어오는 Message 처리의 기본 형식을 ProtocolSession에

서 제공한다. 그림 7에서 보는 것과 같이 모든 패킷은 NIC 객체를 거쳐 IP 클래스를 통과하게 된다. 내부로 들어오는 패킷의 비교할 수 있는 가장 적절한 위치에 있으며 서비스 요청 및 메시지를 처리할 수 있어 IP 클래스를 확장하여 IPS 모듈이 들어간 SIP 클래스를 구현하였다.

IPS Package를 구성하는 클래스는 그림 8과 같이 각각의 기능을 가지고 있다. 외부 네트워크에서 들어오는 패킷을 직접 비교하기 위해서는 규칙이 필요하다. 이 규칙의 내용을 가진 클래스가 RuleDbc 클래스이며, 이 클래스는 초기화 작업도 담당한다.

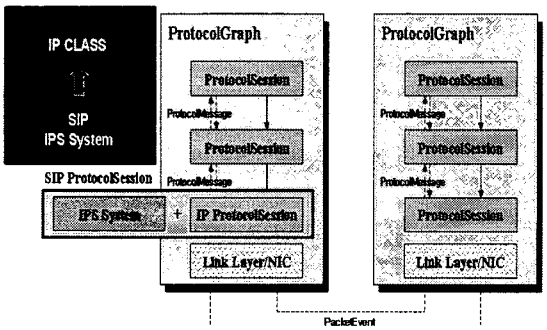


그림 7. IP 클래스를 확장 추가한 IPS 클래스

SelectObject 클래스는 특정 이상한 패킷이 정의된 규칙 집합과 일치하는지 여부를 검사한다. 그리고 이상 패킷의 행동 여부를 결정하는 것은 Selectoptobject 클래스가 하게 된다. 이런 이상 패킷에 대한 로그, 경보, 패킷 통과 여부를 결정하는 것은 AlertDbc 클래스에서 이루어진다.

5. IPS 기능 검증

본 장에서는 패킷 필터링 기능과 실시간 트래픽의 흐름을 탐지할 수 있는 IPS 기능을 검증하기 위해 SSFNet 시뮬레이터를 사용하여 가상 네트워크를 모델링하고, 패킷 조작기를 사용하여 사이버 공격을 했을 경우, 제안한 보안 모듈이 정상적으로 동작이 되는지를 보이고자 한다.

5.1 네트워크 구조

시뮬레이션을 위한 네트워크는 그림 9와 같이 NET0와 NET1 두 개로 나누어 설정하였다. NET0에는 4개의 서버들을 두었고, NET1은 1:0, 1:1, 1:2, 1:3의 네 개의 라우터로 구성하였다. NET0와 NET1를 연결하는 라우터는 2번과 3번에 의해 연결 된다

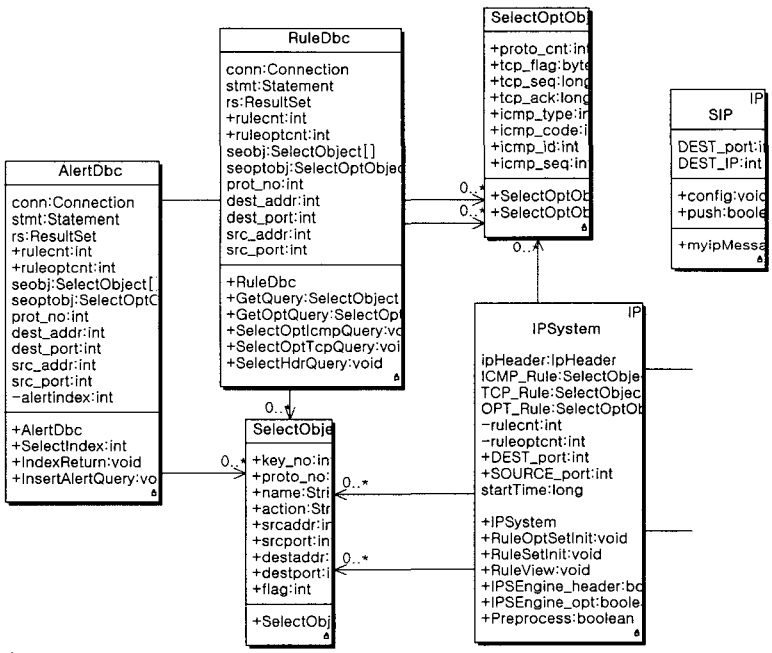


그림 8. IPS 클래스 기본 구조

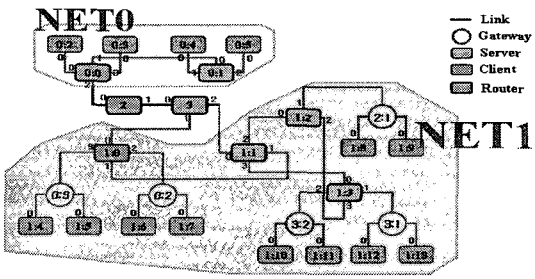


그림 9. 네트워크 구성도

5.2 기능 검증

구현된 IPS의 시뮬레이터의 기능을 검증하기 위해 다음과 같은 내용을 통해 기능을 검증하였다.

SSFNet에서 IPS 설정을 위해서는 네트워크 모델을 기술할 때 SSF.Net Package의 네트워크 구성요소인 Router에 DML로 IPS의 기본 구조를 기술하도록 설계하였다. 그림 10는 라우터에 IPS를 DML로 설정한 것이다.

ProtocolSession은 일반적으로 사용하는 IP가 아닌 제안한 보안 기능을 가진 SIP 클래스를 지정하였다. 각 interface에서는 버퍼의 크기를 설정하여 패킷의 크기가 버퍼의 크기를 넘어갈 수 없도록 설정하였다. 그리고id의 범위를 주어 적용 id의 경우에만 기본적인 규칙이 적용되도록 하였다.

그림 9의 시뮬레이션 네트워크 환경 중에 1:0, 1:1, 1:2 라우터에 IPS 모듈을 설정하였다. 본 논문에서 제안한 IPS는 외부네트워크에서 내부 네트워크로 들어오는 모든 패킷을 실시간 탐지하여 로그화면으로 직접적으로 확인 할 수 있도록 설계하였다.

```
router [
  idrange [from 1 to 2]
  graph [
        ProtocolSession [
                name ip use SIP
        ]
  ]
  interface [
    id 0
    buffer 8000 _extends .dictionary.100BaseT
  ]
  interface [
    idrange [from 1 to 2]
    buffersize 16000
  ]
  route [dest default interface 0]
]
```

그림 10. IPS 설정을 위한 라우터 DML 기술

```
1          1861913663 tcp 10:1600 -> 6:10001
2          1867016863 tcp 10:1600 -> 6:10001
.....
.....
.....
32         1867016863 tcp 13:1600 -> 4:10001
33         1861913663 tcp 13:1600 -> 4:10001
34         1867016863 tcp 12:1600 -> 5:10001
35         1861913663 tcp 12:1600 -> 5:10001
36         1867016863 tcp 17:1600 -> 10:10001
37         1861913663 tcp 17:1600 -> 10:10001
38         1867016863 tcp 27:1600 -> 14:10001
39         1861913663 tcp 27:1600 -> 14:10001
40         1867016863 tcp 24:1600 -> 16:10001
41         1861913663 tcp 24:1600 -> 16:10001
42         1867016863 tcp 18:1600 -> 8:10001
43         1861913663 tcp 18:1600 -> 8:10001
44         1867016863 tcp 11:1600 -> 2:10001
45         1861913663 tcp 11:1600 -> 2:10001
.....
.....
```

그림 11. 라우터 1:1의 로그화면

그림 11은 실시간 탐지 기능을 가진 1:1 라우터의 네트워크 트래픽에 대한 로그 화면을 나타낸 것이다.

본 논문에서는 제안한 IPS는 이상 징후 감지 후 사전에 차단할 수 있는 기능을 가지고 있다. 본 논문에스는 사용자가 명시한 트래픽만을 감시 대상 트래픽으로 여기고 유해 패킷인지 아닌지를 판단하고 있다. 이를 위해 패킷 조작기를 사용하여 패킷 헤더내의 프로토콜 타입 및 패킷의 사이즈를 조절하여 유해 패킷을 생성하여 각 노드로 전송하였다. 그림 12는 라우터 1:2에서 본 패킷 차단내용이다. TCP 프로토콜타입을 사용하여 0.0.0.14 주소를 가진 노드가 0.0.0.10 주소를 가진 노드에게 메시지를 보낼 경우, 신뢰성 있는 TCP는 SYN 메시지에 대해 ACK 메시지를 보내 성공적인 설정이 이루어진다는 것을 화면을 통해 검증할 수 있다. 하지만, 송신지 주소가 0.0.0.6에서 0.0.0.10으로 보내 TCP 메시지는 송신지 주소와 프로토콜 타입에 대한 규칙과 일반적인 패킷의 사이즈에 비해 갑자기 커진 패킷의 크기를 보고 직접 판단하여 차단한 내용을 확인 할 수 있다.

```
** PingClient received (RESPONSE 48 bytes (0/0) from 0.0.0.10) after 0.014849726
seconds
1.6164582O4 TCP host 11 src={0.0.0.14:10001} dest={0.0.0.10:1600} Active Open
1.6187207Z1 TCP host 21 src={0.0.0.10:1600} dest={0.0.0.14:10001} SYN recvd
1.62720324 [ sid 1 start 1.616458204 ] client 11 srv 21(0) rcvd 1000B at 744.529kbps -
read() SUCCESS
1.62720324 TCP host 11 src={0.0.0.14:10001} dest={0.0.0.10:1600} Active Close
1.6294657S7 TCP host 21 src={0.0.0.10:1600} dest={0.0.0.14:10001} Active Close
1.82214099 TCP host 12 src={0.0.0.6:10001} dest={0.0.0.10:1600} Active Open
    Packet Drop by IPS: 6
    Packet Drop by IPS: 6
    Packet Drop by IPS: 6
121.339234331 TCP host 11 src={0.0.0.14:10001} dest={0.0.0.10:1600} 2MSL timeout,
connection closed
121.561760631 TCP host 21 src={0.0.0.10:1600} dest={0.0.0.14:10001} 2MSL timeout,
connection closed
```

그림 12. IPS 시뮬레이션에서 이상 패킷에 대한 차단

6. 결 론

본 논문에서는 대규모 네트워크를 표현하고, 이산 프로세스 기반 사건 중심 시뮬레이션 시스템인 SSFNet에 보안 기능을 추가하였다. 그리고 SSFNet은 패킷을 직접 조작할 수 있는 라이브러리를 제공하지 않고 있다. 그래서 패킷 조작기를 추가하고, 보안 시스템인 방화벽과 IPS를 모델링하여 SSFNet에서 시뮬레이션 할 수 있도록 하였다. 이 보안 구조는 Java로 구현되었으며, DML로 기술하여 사용한다.

기능 검증을 위한 시뮬레이션 환경을 구성하고, 확장된 SSFNet을 사용하여 IPS을 구현하여, 실시간 패킷의 로그화면과 규칙 집합과 일치하는 유해 패킷이 드롭시킨 기능을 확인 할 수 있었다.

참 고 문 헌

[1] 정보보호센터, 정보시스템 침해사고 방지기술 개발, 정보보호센터 논문개발보고서, 1998

[2] 정보통신부고시, 정보통신망 침입차단시스템 평가 기준, 정보통신부, pp. 1-2, 2002.

[3] S. Kumar and E. Spafford, "A pattern-matching model for misuse intrusion detection," *National Computer Security Conference*, Vol. 17, pp. 11-21, 1994.

[4] SSFNet HomePage, *http://www.ssfnet.org.*

[5] SSFNet 2.0 API Documents, *http://www.ssfnet.org/javadoc.*

[6] James H. Cowie, Editor "Scalable Simulation Framework API Reference Manual," *version 1.0, Documentation draft*, 1999.

[7] Paul Reeder, "Intrusion Detection, the next generation," *IEEE*, 2001.

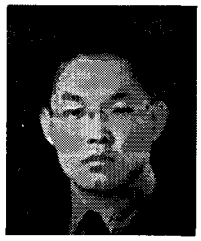
[8] Biswanath Mukherjee, L.Told Heberlein, and Karl N. Levitt "Network Intrusion Detection" *IEEE*, 1994.

[9] SNORT HomePage, *http://www.snort.org.*

[10] R. Durst, T.Champion, B. Witten, E. Miller, and L. Spagnuolo. "Testing and Evaluating Computer Intrusion Detection System," *CACM*, Vol. 7, NO. 42, pp. 53-61, 1999.

[11] Jae-Hyuk Lee, Eul Gyu Im, Joo Beom Yun, and Seung-KyuPark, "Network Intrusion and defense simulation framework based on SSFNet," *International Conference*, Vol. 1, NO. 6, 2004.

[12] 최준호, 김판구, 네트워크상에서의 바이러스 차단을 위한 방화벽시스템의 설계 및 구현, 정보처리 학회 논문이지 C 제8-C권 4호



김 용 탁

1998년 동의대학교 공과대학 컴퓨터공학과 학사
2003년 동의대학교 공과대학 컴퓨터 공학과 석사
2006년 동의대학교 공과대학 컴퓨터 응용공학과 박사

관심분야 : 모바일 프로토콜, 무선 네트워크, 네트워크 보안, 인터넷 QoS

김 태 석

1992년 일본 KEIO대학 이공학부 계산기과학전공 졸업
1992년 일본 KEIO대학 이공학부 객원연구원
1993년~현재 동의대학교 컴퓨터소프트웨어공학과 교수
2000년~2003년 동의대학교 전산정보원장
2000년~2003년 (재)부산테크노파크 운영위원
2003년~2005년 동의대학교 교무처장
관심분야 : 인터넷응용, 원격강의, 자연어처리

권 오 준

1986년 경북대학교 전자공학과 (공학사)
1992년 충남대학교 전산학과(이학석사)
1998년 포항공대 전자계산학과 (공학박사)
1986년~2002년 한국전자통신연구원 선임연구원
2000년~2002년 동의대학교 자연과학대학 컴퓨터통계학과
2002년~현재 동의대학교 공과대학 소프트웨어공학과 부교수
관심분야 : 컴퓨터네트워크, 정보보호, 인공신경망