SSFNet 기반의 침입평가데이터 생성기 설계 및 구현

이영수*, 문길종*, 김용민**, 노봉남***
*전남대학교 정보보호협동과정
**전남대학교 전자상거래전공
***전남대학교 전자컴퓨터정보학부
e-mail : churack@lsrc.jnu.ac.kr

Design and Implementation of Intrusion Evaluation Dataset Generator based on SSFNet

Young-Soo Yi*, Gil-Jong Mun*, Yong-Min Kim**,
Bong-Nam Noh***
*Interdisciplinary Program of Information Security,
Chonnam National University
**Dept. of Electronic Commerce, Chonnam National University
***Div. of Electronics, Computer and Information Eng.,
Chonnam National University

요 약

정보보호 분야에서 네트워크 시뮬레이터에 대한 관심이 커지고 있으나 여러가지 제약 때문에 연구 및 개발이 미흡하다. 특히 침입탐지 시뮬레이터의 평가를 위한 적절한 데이터가 존재하지 않아 침입탐지 시뮬레이터가 적절한지 판단할 근거 자료가 충분하지 않다. 본 논문에서는 네트워크 시뮬레이터에서 DARPA 99 데이터셋을 활용하는 방법으로 트래픽 생성기를 설계 및 구현 하였으며, 그 결과가 정상적으로 동작함을 확인하였다.

1. 서론

네트워크의 규모가 방대해짐에 따라 네트워크 시뮬레이터를 이용한 연구에 관심이 높아지고 있다. 정보보호 분야 역시 네트워크 시뮬레이터에 대한 요구는 많아 졌지만, 방대한 네트워크 표현, 침입과 방어의 복잡성 및 다양성 표현의 어려움으로 연구에 많은 진척이 이루어 지지 않고 있다. 특히 침입탐지 연구를 위해 어플리케이션과 그 취약점을 표현해야 한다는 제약이 있으며, 또한 침입탐지 시뮬레이터에 대한 적절한 평가 데이터셋이 존재하지 않아, 침입탐지 시뮬레이터가 적절한지 판단할 근거 자료가 충분하지 않다.

본 논문에서는 침입탐지 시뮬레이터의 평가데이터셋을 생성하기 위해 DARPA 99 데이터셋[1]을 침입탐지 시뮬레이터에 발생시키는 방법을 제시하고 침입평가데이터 생성기를 구현, 설명한다.

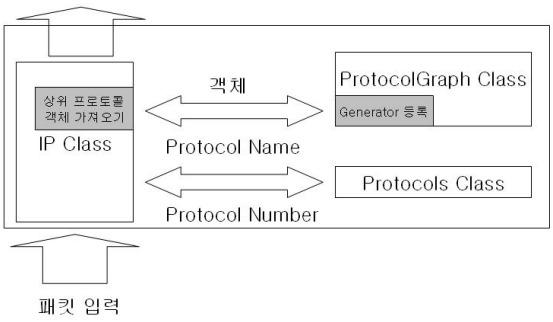
2. SSFNet

NS-Ⅱ[2, 3]와 함께 대표적인 네트워크 시뮬레이터인 SSFNet(Scalable Simulation Framework Network Models)[4, 5]은 DML(Domain Modeling Language)을 이용하여 10만개 이상의 시스템이 존재하는 대규모 네트워크를 호스트 단위로 디자인 할 수 있다. SSFNet이 지원하는 호스트 모델링과 라우팅 프로토콜, 네트워크 속도 등 각 네트워크가 갖는 현실세계의 특성을 바탕으로 네트워크를 구성할 수 있기 때문에 현실 세계와 유사하게 동작하는 네트워크 환경의 모습을 구성하고 관찰할 수 있다. SSFNet은 라우터, 링크 네트워크 인터페이스 카드 등 대부분의 인터넷 서브 시스템들을 시뮬레이션하는 데 필요한 다양한 객체들이 자바로 구현되어 있어 시뮬레이션 특성에 맞추어 그들의 특성을 변경할 수 있다는 장점을 가지고 있다.

3. SSFNet IP 클래스의 확장

트래픽 생성기는 프로토콜에 영향을 받지 않아야 한다. 그러므로 가장 하위 계층에서 구현을 하여야 하며 상위 계층의 모든 프로토콜을 조작 할 수 있어야 한다. SSFNet에서 접근 할 수 있는 가장 하위 계층의 프로토콜은 네트워크계층으로 IP 프로토콜을 기본으로 한다. SSFNet에서 네트워크계층과 전송계층의 통신은 IP 헤더의 프로토콜 번호를 이용한다. 트래픽 생성기의 경우에는 SSFNet에서 제공하는 TCP, UDP, ICMP 프로토콜을 전부 조작 할 수 있어야 한다. 그리고 트래픽 생성기는 특별한 프로토콜 번호를 가지고 있지 않기 때문에 트래픽 생성기를 구현하기 위해서는 *IP* 클래스를 확장해야 한다.

SSFNet은 네트워크계층에서 전송계층으로 데이터를 전송하기 위해서 프로토콜 번호와 이름을 이용하여 객체를 찾는 방법을 사용하고 있다. 이 방법은 트래픽 생성기를 이용하여 TCP 또는 UDP 데이터를 전송할 때 응답 패킷을 트래픽 생성기가 아닌 TCP 또는 UDP 프로토콜로 패킷을 보내기 때문에 패킷이 올바르게 전송되지 못하게 된다. 이 문제를 해결하기 위해 SSFNet의 *ProtocolGraph* 클래스와 *IP* 클래스를 수정해야 한다. 특히 트래픽 생성기는 특정 프로토콜이 아닌 모든 프로토콜을 캡처할 수 있어야 하기 때문에 *ProtocolGraph* 클래스에 트래픽 생성기를 등록해 줘야 한다. 또한 *IP* 클래스에 상위 프로토콜의 객체를 가져오는 부분을 수정해 주어야 한다. SSFNet 클래스 확장 모습은 (그림 1)과 같다. *IP* 클래스는 프로토콜 번호를 이용하여 프로토콜에 대한 이름을 가져오고 그것을 이용하여 *ProtocolGraph*에서 상위 계층의 객체를 가져온다. 수정된 부분은 프로토콜 이름을 이용하여 객체를 검색하기 전에 *Generator* 객체가 존재하는지 확인을 하는 것이다. *Generator* 객체가 존재하면 *Generator* 객체를 반환하게 되며, 존재하지 않으면 프로토콜 이름에 따른 상위 프로토콜의 객체를 반환하게 된다.



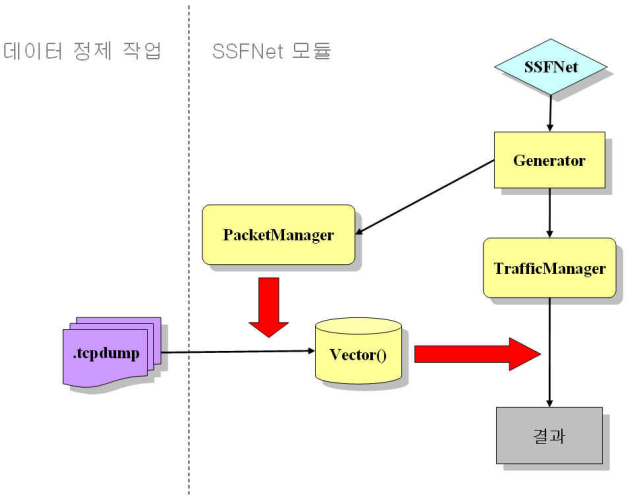
(그림 1) SSFNet 클래스 확장

4. 평가데이터 트래픽 생성기 설계

본 논문에서는 트래픽 생성을 위해 SSFNet을 이용하여 트래픽 생성기(Traffic Generator)를 구현하였다. 트래픽 생성기는 실세계의 TCP 덤프 데이터를 바탕으로 가상 시뮬레이터에서 똑같은 패킷을 생성하는 역할을 한다. 트래픽 생성기를 이용하면 제한된 네트워크에서 발생한 TCP 덤프 데이터를 가상 시뮬레이터에 구축한 임의의 네트워크 토폴로지에서 발생시킬 수 있다.

SSFNet에 트래픽 생성기를 만들기 위해서는 호스트에 대한 기본 설정을 하는 부분과 각 세션을 담당하는 부분, 그리고 실세계의 TCP 덤프 데이터를 읽어 트래픽 정보를 저장하고 있는 부분이 필요하다. (그림 2)는 트래픽 생성기의 흐름도이다.

SSFNet이 실행이 되고 DML의 *ProtocolSession*에 트래픽 생성기가 등록이 되어 있으면 *Generator* 클래스가 실행이 되게 된다. *Generator* 클래스는 *PacketManager* 클래스와 *TrafficManaager* 클래스를 생성하게 된다. *PacketManager* 클래스의 실행을 위해서는 패킷 헤더에 대한 정보가 필요하다. 이 정보는 TCP 바이너리로부터 가져오게 된다. *PacketManager* 클래스는 헤더 정보 값들을 이용하여 패킷들의 정보를 벡터로 가지고 있으며, *TrafficManager*는 이 벡터 값들을 이용하여 SSFnet의 가상 시뮬레이터 상에 데이터를 생성하게 된다.



(그림 2) 트래픽 생성기의 흐름도

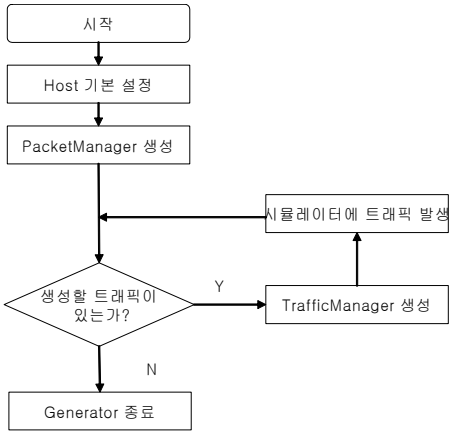
4.1 Generator 클래스

Generator 클래스는 SSFNet의 *ProtocolSession* 클래스를 상속 받는다. 트래픽 생성기의 디버그 설정 등과 같은 일반적인 설정을 담당하며 각 트래픽 세션을 생성한다. *Generator* 클래스의 동작 절차는 (그림 3)과 같다.

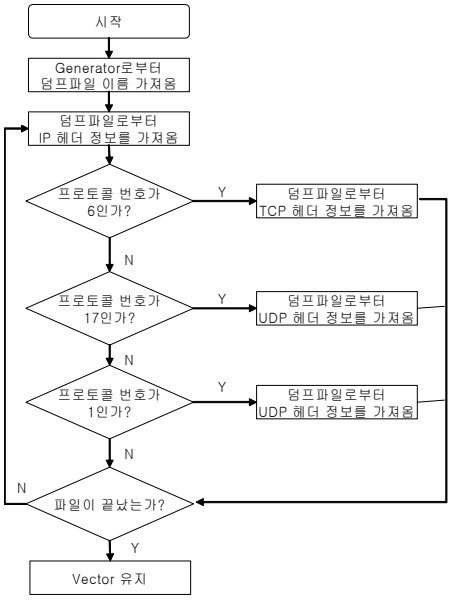
Generator 클래스는 DML 파일로부터 트래픽 생성기에 대한 기본 설정을 읽고 설정 작업을 한 후 실행시킨다. *Generator* 클래스는 *PacketManager* 객체를 생성하여 자신이 생성해야 하는 트래픽 정보를 얻게 된다. *Generator* 클래스는 *PacketManager* 클래스의 벡터를 검사해서 생성할 트래픽이 있으면 *TrafficManager* 객체를 생성해 트래픽 세션을 담당하도록 한다. 생성 된 *TrafficManager* 클래스는 각 세션의 트래픽을 전송 한다.

4.2 PacketManager 클래스

PacketManager 클래스는 트래픽 벡터를 생성하기 위해 TCP 바이너리 파일로부터 헤더 정보를 추출하고 그 정보를 유지한다. 벡터는 네트워크계층과 전송계층 정보로 나누어져 있다. (그림 4)는 *PacketManager* 클래스의 동작 절차를 표현한 것이다. *Generator* 클래스가 *PacketManager*를 생성할 때 생성자의 인자 값으로 DML에 명시된 TCP 바이너리 파일의 이름을 넘겨주게 된다. *PacketManager* 클래스는 TCP 바이너리 파일을 열고 가장 먼저 네트워크계층 정보를 추출하여 벡터에 저장한다. 전송계층의 정보는 네트워크계층의 프로토콜 번호에 따라 TCP, UDP, ICMP 헤더 정보로 추출된다. SSFNet에서는 TCP, UDP, ICMP 프로토콜을 지원하고 *PacketManager* 클래스도 TCP, UDP,



(그림 3) *Generator* 흐름도



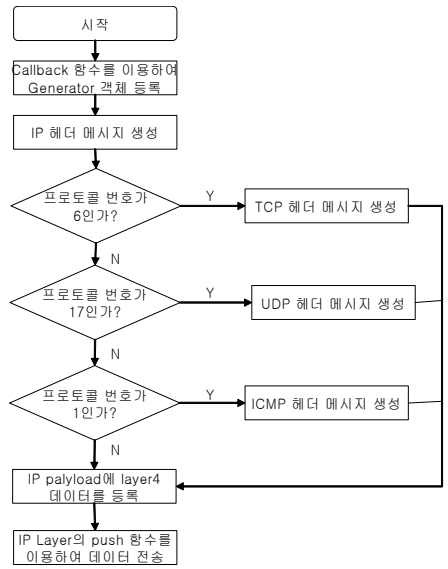
(그림 4) *PacketManager* 흐름도

ICMP 패킷을 벡터에 저장할 수 있다. 마찬가지로 *TrafficGenerator* 모듈도 TCP, UDP, ICMP 트래픽을 컨트롤 할 수 있다.

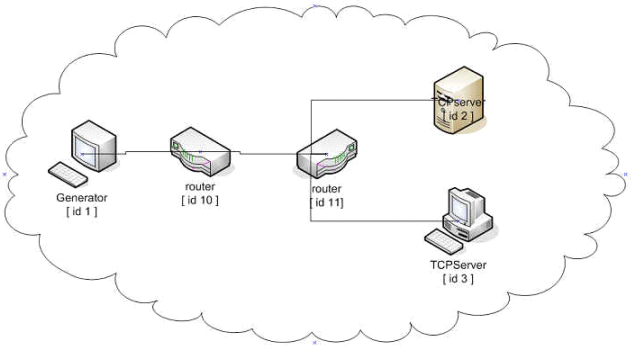
4.3 TrafficManager 클래스

TrafficManager 클래스는 SSFNet의 *ProtocolSession* 클래스를 상속 받고, push() 함수를 이용하여 *IP* 클래스와 직접적인 통신을 맡게 된다. (그림 5)는 *TrafficManager* 클래스의 동작 절차를 보여 준다.

TrafficManager 클래스는 시작할 때, *Generator* 객체를 자신에게 등록한다. 그 후 IP 헤더 메시지를



(그림 5) *TrafficManager* 흐름도



(그림 6) 네트워트 구성도

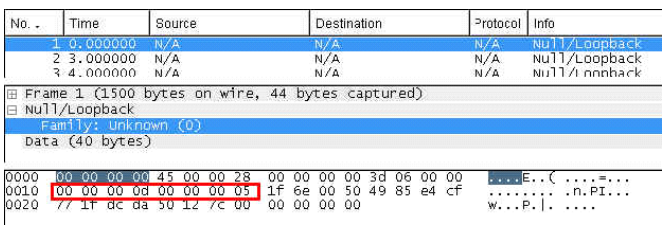
생성하고 IP 헤더의 프로토콜 번호에 따라 전송계층의 데이터를 생성하게 된다. 전송계층 데이터 생성 후 IP 헤더 메시지의 페이로드 값으로 전송 계층의 데이터를 등록시켜 준다. 그리고 *IP* 클래스의 push() 함수를 이용하여 SSFNet 가상 시뮬레이터 상에 패킷을 전송하게 된다. 패킷 전송을 완료 시킨 후에 응답을 기다리는 동안 가상 시뮬레이터가 블록되는 것을 막기 위해 등록했던 *Generator* 객체를 호출하게 된다.

5. 테스트

트래픽 생성기가 올바르게 동작하는지 확인을 하기 위해 최소의 네트워크를 구성한 후 동작 실험을 하였다. (그림 6)은 테스트를 위한 네트워크 구성도이다.

간단하게 한 쌍의 TCP 서버와 TCP 클라이언트, 그리고 두 대의 라우터와 패킷을 생성하는 트래픽 생성기로 구성되어 있다. 트래픽 생성기는 DARPA 99 데이터셋에서 2주차 화요일 공격중 하나를 이용하였다. 이 공격은 총 17개의 TCP 패킷을 생성하게 된다. SSFNet의 tcpdump 기능을 이용하여 TCP 서버에서의 트래픽을 캡처해 보았다.

(그림 7)에서 박스 안의 내용이 IP 헤더의 출발지 주소와 목적지 주소이다. 그리고 그 주소의 내역은 (그림 8)에서 NHI[3]로 확인 할 수가 있다. 즉 IP



(그림 7) TCP 서버의 덤프 파일

CIDR	IP Block	b16	NHI
--	0.0.0.0/27	0x00000000	
0	0.0.0.12/30	0x0000000c	1(0) 10(0)
1	0.0.0.8/30	0x00000008	2(0) 11(0)
2	0.0.0.4/30	0x00000004	3(0) 11(2)
3	0.0.0.0/30	0x00000000	10(1) 11(1)

NHI Addr	CIDR Level	IP Address Block	% util
--	--	0.0.0.0/27	56.25

(그림 8) NHI 주소

'0.0.0.13'은 NHI 1(0)인 트래픽 생성기를 의미하며 IP '0.0.0.5'는 NHI 3(0)인 TCP 서버를 의미한다. 트래픽 생성기로부터 TCP 서버까지 패킷이 올바르게 전송되었다는 것을 확인 할 수 있다.

6. 결론 및 향후과제

본 논문에서는 트래픽 생성기 모듈 구현으로 SSFNet을 확장하여 임의의 네트워크 토폴로지에서 침입평가 데이터 트래픽을 정상적으로 생성하는 것을 확인하였다. 그러나 현실 네트워크와 시뮬레이터의 결과가 동일하다고 할 수는 없다.

향후에는 연구 결과를 더 발전시키기 위해 DARPA 99 데이터셋과 시뮬레이터에 발생시킨 침입평가 데이터와 차이점을 비교 분석할 예정이며 SSFNet에서 구현이 되지 않은 TCP 헤더의 필드의 영향에 대해 조사하고자 한다.

참고문헌

[1] Richard P.Lippmann and David J. Freid etc. "Evaluating Intrusion Detection System:The 1998 DARPA off-line Intrusion Detection Evalution"

[2] NS-Ⅱ "http://www.isi.edu/nsnam/ns/"

[3] Richard Blum "Network Performance Open Source Toolkit" WILEY 2003

[4] 유관종 외 3인 "사이버 침입 탐지 시뮬레이션을 위한 SSFNet 기반 IDS의 확장" 한국정보과학회 가을 학술발표논문집 Vol.32, No.2 2004

[5] James Cowie, Hongbo Liu, Jason Liu, David Nicol, Ansdy Ogielski "Towards Realistic Million-Node Internet Simulations" International Conference on Parallel and Distributed Processing Techniques and Applications 1999