네트워크 방어 시뮬레이터 설계 및 구현

정회원 이 철 원*, 윤 주 범**, 임 을 규***

Design and Implementation of Network Defense Simulator

Cheol Won Lee*, Joo Beom Yun**, Eul Gyu Im*** *Regular Members*

요 약

최근 정보보안에 대한 관심이 고조되면서 사이버 침입 및 방어 연구를 위한 정보보호 시뮬레이터 개발이 요구되고 있다. 지금까지 정보보호 시뮬레이션은 소규모 네트워크의 보안도 평가나 정보보호 시스템의 성능 평가가 주 목적이었다. 하지만 최근에 대규모 네트워크를 대상으로 하는 사이버 테러가 자주 발생함에 따라 대규모 네트워크의 시뮬레이션이 가능한 시뮬레이터가 필요하게 되었다. 이에 본 논문에서는 실세계의 인터넷 망과 유사한 대규모 네트워크를 구성하고 그 위에서 사이버 침입 시나리오를 수행하는 시뮬레이터의 설계 및 구현 방법을 제안하고 있다. 제안한 네트워크 방어 시뮬레이터는 SSFNet 프로그램을 기반으로 클라이언트-서버 구조로 구현되었다. 사이버 침입 시나리오는 수행 순서를 표시할 수 있는 개선된 공격 트리 모델을 이용하여 표현하였고 시뮬레이션 수행 과정을 시각적으로 보여주기 위하여 그래픽 사용자 인터페이스를 제공하였다. 또한 네트워크 방어 시뮬레이터의 유효성을 살펴보기 위해 시뮬레이션 수행 결과를 분석하였다.

Key Words : SSFNet; Scenario; Network Defense; Simulation Client; Simulation Server.

ABSTRACT

Information security simulator is required for the study on the cyber intrusion and defense as information security has been increasingly popular. Until now, the main purposes of information security simulation are security estimation of small network as well as performance analysis of information protection systems. However, network simulators that can simulate attacks in a huge network are in needs since large scale internet attacks are very common in these days. In this paper we proposed a simulator design and its implementation details. Our simulator is implemented by expanding the SSFNet program to the client-server architecture. A cyber attack scenario used in our simulator is composed by the advanced attack tree model. We analyzed the simulation results to show the correctness of our network defense simulator.

Ⅰ. 서 론

네트워크에 대한 사이버 침입과 이에 따른 네트워크 행동 변화 및 파급효과에 대한 연구는 실세계에서 발생하는 사이버 침입과 방어를 위한 기반 연구이다. 그러나, 네트워크의 방대함, 침입과 방어의 복잡성 및 다양성 표현 어려움 등으로 인하여 아직까지 많은 연구의 진전이 없는 분야이기도 하다. 특정 침입에 대한 네트워크 행동 변화 및 방어 기법에 대한 자료를 얻기 위하여, 실제 네트워크에 사이버 침입을 시도하고 그때의 네트워크 행동 변화를 관찰하는 것이 가장 좋은 방법이다. 그러나 발생 가능한 모든 침입을 시도하여 필요한 자료를 얻는 것은 위험성이 존재할 뿐만 아니라 현실적으로 여러 가지 제약사항이 따른다. 좋은 대안으로 시뮬레이션 기법을 들 수 있으며, 이를 수행하는 정보보호 시뮬레이터의 개발 노력은 계속되어 왔다. 첫째로, 개념이나 이론을 테스트하기 위하여 시뮬레이션을 사용

* 국가보안기술연구소(cheolee@etri.re.kr), ** 국가보안기술연구소(netair@etri.re.kr) *** 국가보안기술연구소(imeg@etri.re.kr)
논문번호 : 030182-0429, 접수일자 : 2003년 4월 17일

하였다. 예를 들어, Smith와 Bhattac Harya[1]는 소규모 네트워크에서 침입차단시스템 위치에 따른 성능을 평가하기 위해 시뮬레이션을 사용했다. 시뮬레이션은 또한 네트워크 형태(topology)와 성능을 평가하기 위해 사용되었다[2,3]. 그러나, 본격적인 네트워크 방어 시뮬레이션은 Mostow, et al.[4]의 IAS(Internet Attack Simulator)에서 시작되었다. IAS에서는 3가지 공격 시나리오(서비스 거부 공격, 허가되지 않은 접근, 속임(spoofing))를 시뮬레이션하였다. Donald Welch, et al.[5]의 정보전 시뮬레이션 프레임워크에서는 암호 가로채기(password sniffing), 침입차단시스템 효과에 관한 시뮬레이션을 수행하였다. 그러나 이와 같은 네트워크 방어 시뮬레이터들이 현재 실세계 네트워크와 사이버 침입을 제대로 표현하는 수준은 아니라고 판단된다.

가상 침입을 수행하고 이에 따른 네트워크의 행동 변화를 시뮬레이션하기 위해서는 다음과 같은 요건들이 만족되어야 한다. 첫째, 시뮬레이션의 대상이 되는 네트워크의 실제 모습을 모델링할 수 있어야 한다. 특히 세계적인 규모의 방대한 인터넷을 모델링할 수 있어야 하며, 모델링된 네트워크상에서 시뮬레이션이 가능하여야 한다. 기존 연구[4,5]의 경우 노드 수가 50개 이하인 소규모 네트워크를 구성하여 시뮬레이션을 수행하였으나 본 논문에서는 SSFNet(Scalable Simulation Framework Network Models)[6]을 시뮬레이션 엔진으로 사용하여 최대 100,000개의 노드를 표현할 수 있다. 둘째, 네트워크를 구성하는 시스템들의 특성을 네트워크 모델에 반영할 수 있어야 한다. 본 논문의 시뮬레이터에서는 통신 프로토콜 스택 상에서 일어나는 다양한 패킷의 처리절차를 표현할 수 있으며 네트워크 전송속도 및 대역폭에 관한 특성을 표현하고 있다. 셋째로 다양한 사이버 침입의 특성에 가깝도록 가상 침입의 시뮬레이션이 가능하여야 한다. 본 논문의 시뮬레이터는 하나의 시뮬레이션 프로세스에 의해 모든 동작이 실행되는 것처럼 시뮬레이션하기보다는 시뮬레이션 대상 노드 당 하나의 스레드가 존재하여 스레드 사이의 통신으로 호스트들간의 통신을 묘사한다. 넷째로는 네트워크 침입을 방어하는 시스템들의 특성을 적용할 수 있어야 한다. 이를 위해 본 논문의 시뮬레이터는 침입차단시스템 및 침입탐지시스템을 실제 시스템과 유사하게 표현하였다. 다섯째로는 가능한 많은 사이버 침입 시나리오를 작성할 수 있도록 하는 네트워크 방어 시나리오 편집기가 필요하다. 우리의 시뮬레이터는 버퍼 오버플로우 침입을 이용한 관리자 권한 획득, 분산서비스 거부, 웜 유포에 대한 시나리오 편집기를 구현하였다. 지금까지 이러한 요건을 모두 만족하는 시뮬레이터는 존재하지 않았으며 한정된 분야에 대한 시뮬레이션을 수행하는 연구가 전부였다. 이에 본 논문에서는 위에서 기술한 기준을 모두 만족하기 위하여 대규모 네트워크에서의 침입과 이에 따른 네트워크 행동 시뮬레이션을 위해 SSFNet을 확장, 구현하였으며 다양한 구조로 사이버 침입 및 방어 시뮬레이션을 수행할 수 있는 방안을 제시한다.

본 논문은 다음과 같이 구성된다. 2장에서는 기존 연구 내용과 새로운 네트워크 방어 시뮬레이터의 프레임워크를 제시하고, 우리가 구현한 네트워크 방어 시뮬레이터의 전체 구조에 대해 기술한다. 3장에서는 우리의 시뮬레이터에서 구현한 시나리오 중 분산 서비스 거부 공격을 시뮬레이션한 결과와 실제 환경을 구성하여 측정한 결과를 비교한다. 4장에서는 결론 및 향후 과제를 제시한다.

Ⅱ. 시뮬레이터 시스템 프레임워크

1. 기존 네트워크 방어 시뮬레이터

Donald Welch, et al.[5]의 네트워크 방어 시뮬레이션 모델은 객체지향모델이다. 다음과 같이 4가지 객체 종류를 가지고 개체(entity)를 모델링하고 각 개체는 속성(attribute)과 행위(behavior)를 가진다.

1. 노드 주소(Node Address)
2. 접속(Connection)
3. 상호작용(Interaction)
4. 정보원자(Infotron)

노드 주소 객체는 워크스테이션, 서버, 라우터 등과 같이 정보를 보내고 받는 네트워크의 노드를 식별하는데 사용하는 객체이다. 속성은 노드의 여러 가지 기본 요소들(프로세서 속도, 메모리 용량, 제공 서비스 등)을 포함하며 행위는 노드가 취할 수 있는 행동(action)을 나타낸다. 접속 객체는 노드 객체 사이의 물리적인 링크(link)를 나타낸다. 접속 객체가 갖는 속성으로는 속도, 연결 노드 정보, 안전성(reliability) 등의 성질을 갖는다. 상호작용(Interaction) 객체는 노드들 사이의 정보의 교환을 나타내는 객체이다. Donald Welch, et al.[5]의 시뮬레이션에서는 상호작용 객체가 패킷 수준을 표현하기 보다는 중요한 이벤트만을 표현하였다. 정보원자

(Infotron)는 시뮬레이션에서 관심이 되는 가장 작은 단위의 정보이다. 이것은 시뮬레이션의 관심에 따라 하나의 전체 데이터베이스가 될 수도 있고, 데이터베이스의 특정 부분만 될 수도 있다. 이와 같은 시뮬레이션 프레임워크로 암호 가로채기(password sniffing)를 모델링을 한 그림이 그림 1 이다.

그림 1의 왼쪽 그림은 암호 가로채기(password sniffing) 시나리오를 나타낸 그림이다. 오른쪽 그림은 시뮬레이션 실행 초기 화면이다. 각 객체들이 클래스 이름과 함께 나타나 있다. 네트워크 방어 시나리오를 왼쪽 그림과 같이 그래프로 표현할 경우 복잡도가 증가하고 실제 시뮬레이션 수행 시 이벤트들 사이의 순서 관계를 알기가 어렵다는 단점이 존재한다. 또한 오른쪽 시뮬레이션 화면의 상호작용은 너무 추상적이어서 실제 시스템의 세밀한 행위를 표현하지 못하고 있다. 예를 들어, 가로채기(Sniff) 행위가 Donald Welch, et al. 시뮬레이션[5]에서는 그냥 하나의 스텝으로 표현되고 화면에 보이는 동작도 없다. 하지만 우리가 개발한 시뮬레이터는 가로채기(Sniff)를 가로채기(sniffing) 프로그램 시작, 네트워크 패킷 가로채기(sniffing), 신용카드 정보 획득 등의 단계로 나누고 패킷이 오가는 상호작용(Interaction)은 빨간 색 선으로 표시한다.

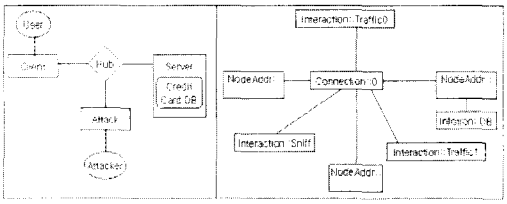


그림 1. Donald Welch, et al.의 시뮬레이션

2. 제안 네트워크 방어 시뮬레이터

1 Sniffing 시나리오 (Scenario)
  1.1 Sniffing 시작 (SequentialAND)
    1.1.1 Attacker PC 연결 (Actor)
    1.1.2 Sniffing 프로그램 시작 (Actor)
  1.2 네트워크 패킷 sniffing (SequentialAND)
    1.2.1 Hub 지나가는 패킷 모두 sniffing (Actor)
  1.3 신용카드 정보 획득 (SequentialAND)
    1.3.1 User 서버 접속 시도 sniffing (Actor)
1.3.2 신용카드 정보 sniffing (Actor)

그림 2. 신용카드 정보 가로채기 시나리오

우리는 공격 트리(Attack Tree)[8]를 변형하여 시뮬레이션 시나리오를 표현한다. 공격 트리는 침입의 최종목표, 즉 마지막으로 수행되는 노드를 루트 노드로 구성하고, 그 루트 노드에 침입을 수행하기 위해 선행되어야 하는 침입들을 하위 노드로 구성한다. 우리는 이 공격 트리에 실행 순서 개념을 부가하여 실행 순서에 따라 트리를 구성하였다. 먼저 공격 트리의 모든 노드는 계층적 레벨과 실행 순서에 따라 단계번호를 부여한다. 단계번호는 그림 2와 같이 구성된다. 트리의 노드에 단계 번호를 부여하여 표현하면, 노드의 트리 구조상의 위치를 확인하기 쉽고, 또한 시뮬레이션 과정에서 노드의 실행순서가 명확해진다. 공격 트리를 구성하는 노드는 기능에 따라 크게 두 가지로 구분된다. 액터 노드(Actor Node)는 하위노드를 가지지 않는 트리의 종단노드로서, 실제 이벤트를 수행한다. 노드의 이름은 이벤트의 이름으로 표기한다. 연산 노드(Operation Node)는 반드시 하나 이상의 액터 노드나 다른 연산 노드를 하위 노드로 가지며, 하위 노드의 실행순서를 지정하고, AND와 OR 같은 논리연산의 의미를 가진다. 이러한 연산 노드는 네트워크 방어 시나리오 작성시 액터 노드들 사이의 관계를 보다 잘 표현할 수 있도록 도와주고, 공통된 이벤트 특성을 갖는 액터 노드를 묶어서 표현할 수 있도록 해주며, 보다 다양한 형태의 침입 시나리오를 구성할 수 있게 해준다. 연산 노드에는 순차(Sequential) AND, 순차(Sequential) OR, 병렬(Parallel) AND, 병렬(Parallel) OR가 존재한다.

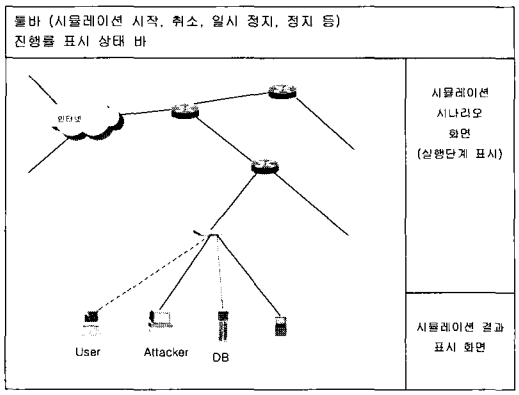


그림 3. 시뮬레이션 실행 화면 구성도

그림 3은 제안하는 네트워크 방어 시뮬레이션 실행 화면이다. 우리는 네트워크를 실제 모습에 가깝

게 모델링 할 수 있도록 그래픽 사용자 인터페이스를 이용하여 네트워크 구성도를 편집할 수 있으며, 이 네트워크 위에서 우측 상단의 시뮬레이션 시나리오를 표현하게 된다. 그리고 우측 하단에 시뮬레이션 결과를 표시하는 화면이 존재한다. 그림 3에서는 신용카드 정보 가로채기 상황 화면을 보이고 있고 중요 신용정보가 오가는 모습을 빨간색 선으로 표시한다. 그림 3에서 보듯이 제안한 시뮬레이터는 시뮬레이션 실행 과정을 그래픽하게 보여 주며, 100,000개의 노드 시뮬레이션도 지원한다. 또한 시뮬레이션 시나리오도 같은 화면에 표시해 줌으로써 시뮬레이션 실행에 대한 이해도를 높이고 시뮬레이션 결과도 같은 화면에 표시하게 된다. 여기서 나타난 객체들은 모두 SSFNet의 클래스(class)로 구현되어 객체지향적으로 동작하게 된다.

제안한 네트워크 방어 시뮬레이터는 그림 4와 같이 클라이언트-서버 모델을 바탕으로 설계되었다. 이 모델을 채택한 이유는 성능 향상을 위한 병렬 및 분산 시뮬레이션이 가능하기 때문이다. 시뮬레이션 클라이언트는 네트워크 방어 시나리오를 작성하고 시뮬레이션 결과를 보여주는 그래픽 사용자 인터페이스 프로그램이다. 시뮬레이션 서버는 시뮬레이션 엔진을 가지고 있으며, 시나리오를 DML(Domain Modeling Language)[7]로 바꾸어 주는 DML 변환기, 시뮬레이션 클라이언트와 통신하기 위한 데몬 등 시뮬레이션 수행에 필요한 핵심 요소들을 가지고 있다.

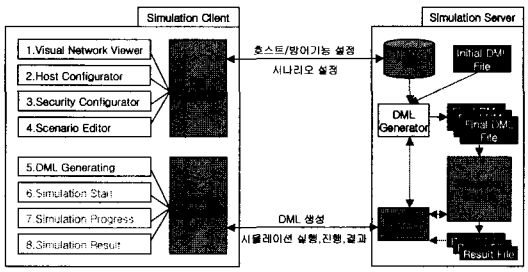


그림 4. 네트워크 방어 시뮬레이터 구조

사용자는 시뮬레이션 클라이언트 프로그램을 실행시킨 후 시뮬레이션 클라이언트 프로그램에서 네트워크를 편집한다. 네트워크 편집이란 네트워크 장비(라우터, 스위치 등)들을 배치하고 호스트 등을 실선으로 연결하여 원하는 형태(topology)로 구성하는 것이다. 네트워크를 구성한 후에는 네트워크 안에 존재하는 호스트의 속성을 설정한다. 그후 시뮬레이션을 수행하고자 하는 침입 시나리오를 편집한다. 시나리오 생성은 단위 시나리오(actor)들을 조합하여 이루어지는데, 시나리오를 시뮬레이션 엔진이 이해하도록 하기 위해서 DML로 변환된다. 지금까지 사용자가 설정한 정보는 데이터베이스 커넥터를 통해 시뮬레이션 서버에 전달된다. 그 후 사용자가 DML 생성을 지시하면 그 요구가 서버로 전달되고, DML 생성기는 네트워크 및 호스트 정보와 시나리오 정보를 합하여 시뮬레이션 엔진이 이해할 수 있는 최종 DML 파일을 생성한다. 그 후 클라이언트의 시뮬레이션 시작 명령에 의해 서버에서는 시뮬레이션이 수행되고 수행된 결과들은 클라이언트로 보내져서 그래픽하게 보여준다.

Ⅲ. 시뮬레이션 결과

이 장에서는 분산 서비스 거부공격 시나리오를 보여 주고 시뮬레이션 수행 결과와 실제 네트워크를 구성하여 측정한 데이터를 비교·분석하였다.

1. 실험 및 시뮬레이션 환경

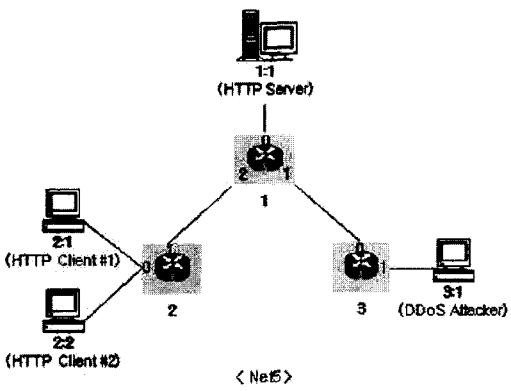


그림 5. 네트워크 구성 환경

그림 5는 분산 서비스 거부 공격에 대한 실제 실험과 시뮬레이터를 사용한 시뮬레이션의 비교를 수행할 네트워크의 모습이다. 호스트 2:1과 2:2는 HTTP 클라이언트이며, 두 개의 HTTP 클라이언트가 HTTP 서버인 호스트 1:1에 HTTP 메시지를 주고받는 도중에 호스트 3:1이 분산 서비스 거부 공격 유형 중에 하나인 ICMP Flooding Attack을 실시한다. 이러한 실험과정에 대해 실제 실험과 시뮬레이터에서 호스트 1:1로 들어가는 단위 시간당 ICMP 패킷의 개수 변화를 각각 측정하여 비교한다.

실험에 사용되는 호스트들은 모두 Pentium III 700MHz, 128MB 메모리를 가진 사양에 Linux 6.0 운영체제를 가지고 있고 네트워크는 10Mbps 환경이다. HTTP 서버는 Apache, HTTP 클라이언트는 SPECweb99, 분산 서비스 거부 공격 도구는 TFN2K를 사용했다.

2. 시뮬레이션 시나리오

실제 실험에서는 호스트 3:1 에서 호스트 1:1 로 TFN2K 도구를 이용하여 분산 서비스 거부 공격을 수행한다. 그러나 우리의 시뮬레이션에서는 어떤 공격을 하기 전에 NetworkScan 등을 해야 하므로 다음의 시나리오대로 수행해야 한다.

```
1. Scenario
1.1 SequentialAND (SequentialAND)
1.1.1 NetworkScan
1.1.2 PortScan
1.1.3 DDoSAgentListen
1.1.4 DDoSMasterListen
1.1.5 AddDDoSAgentToMaster
1.1.6 DDoSInvoker
```

그림 6. DDoS 공격을 위한 시나리오 구성

3. 실험 결과

3.1 실험 데이터

본 논문에서 실제 실험과 시뮬레이션 실행시 ICMP 패킷만 고려하고 TCP나 UDP 패킷은 무시하기로 한다. 실제 실험에서는 대상 시스템이 존재하는 허브에서 대상 호스트에 유입되는 패킷 개수를 관찰하였다. 그림 7을 보면 실제 ICMP Flooding 공격 시간인 12초에서 42초 구간에서 ICMP 패킷의 개수가 현격히 증가하는 것을 볼 수 있다. 공격 전에 0개 였던 ICMP 패킷의 수는 공격 구간에서 초당 평균 8000개 이상까지 증가하는 것을 볼 수 있다.

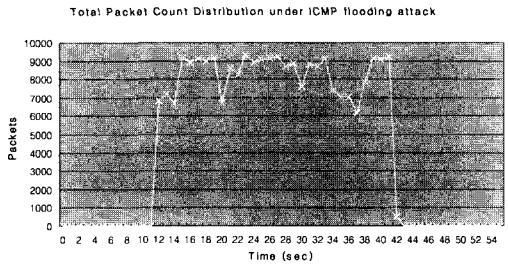


그림 7. 실제 환경에서의 DDoS 공격 결과

3.2 시뮬레이션 결과

분산 서비스 거부 공격 시뮬레이션에서는 호스트 1:1에서 측정한 것이 아니라 침입차단시스템에서 1:1로 들어가는 ICMP 패킷의 수를 측정하였다. 시뮬레이션에서는 실제 분산 서비스 거부 공격이 이뤄지는 23초 이후에 급격한 패킷 증가를 관측할 수 있었다. 공격 이전에 0개 였던 ICMP 패킷의 수는 44초 이후 평균 2500개로 급격히 증가하였다. 그림 8은 분산 서비스 거부 공격의 간격을 0.0004초로 한 그림이다. 초당 대략 2500개의 패킷이 전송됨을 볼 수 있다. 그림 9는 분산 서비스 거부 공격의 간격을 0.00012초로 한 결과이다. 간격을 0.00012초로 함으로써 대략 실제 실험과 비슷한 패킷량을 전송함을 발견하였다.

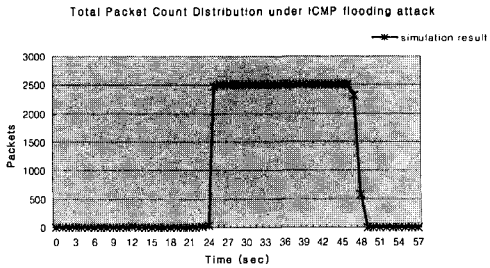


그림 8. 시뮬레이션 환경에서의 DDoS 공격 결과 (간격 0.0004초)

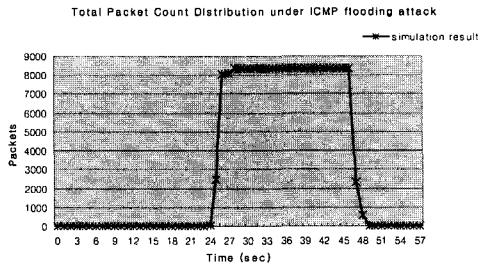


그림 9. 시뮬레이션 환경에서의 DDoS 공격 결과 (간격 0.00012초)

3.3 실험 결과 비교

실험에 대한 결과 비교는 그림10과 같다. 실제 실험에서는 12초 이후부터 ICMP 패킷의 수가 급격

히 증가하고 시뮬레이션에서는 23초 이후에 급격한 ICMP 패킷의 수가 증가하는 것을 살펴볼 수 있다. 시뮬레이션의 결과가 실험 결과보다 더 늦게 패킷이 증가하는 것은 시뮬레이션에서는 NetworkScan 및 PortScan 등의 부가 작업을 실시하기 때문이다. 실제 실험 결과(초당 9000개)와 시뮬레이션 결과(초당 2500개)에서 초당 패킷 개수가 차이나는 것은 실제 실험에서의 패킷이 시뮬레이션보다 좀더 자주 공격 패킷을 보내기 때문이다. 즉, ICMP Flooding의 공격 간격의 차이 때문으로 풀이된다. 마지막으로 실측 결과에서 43초 이후 다시 공격 이전의 상태로 복귀되는 것과는 달리 시뮬레이션 결과에서는 23초부터 66초까지 평균 2500개의 패킷 수가 일정하게 나타났다. 이는 구현된 분산 서비스 거부 공격의 패킷들이 실제 네트워크에서 보다 시뮬레이션 환경의 delay로 인해 좀더 네트워크에 오래 머물기 때문으로 생각된다.

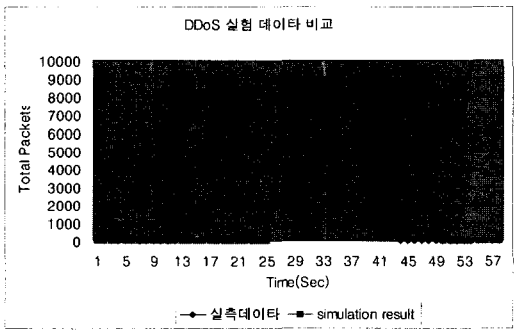


그림 10. DDoS 공격 결과 비교

Ⅳ. 결론 및 향후 과제

SSF(Scalable Simulation Framework)[9]와 SSFNet은 대규모 네트워크를 표현하고 그 위에서 네트워크 행동을 시뮬레이션 하기에 적합한 구조를 제공하고 있다. 이에 본 논문에서는 SSF 및 SSFNet 을 확장, 구현하여 네트워크 방어 시뮬레이터를 개발하였다. 이것은 서버-클라이언트 구조로 구현되었으며 사용자 인터페이스를 담당하는 시뮬레이션 클라이언트와 시뮬레이션을 수행하는 시뮬레이션 서버로 이루어져 있다. 본 논문의 시뮬레이터는 기존 연구들의 한계인 대규모 네트워크를 모델링하고 그 위에서 분산 서비스 거부 공격 및 웜 유포 등을 표현함으로써 최신의 사이버 테러까지도 표현할 수 있다. 이를 위해 대규모 네트워크 설정이 가능하도록 하는 기능과 네트워크 침입 시나리오를 표현하는 기능 모듈도 구현 완료 하였다. 사이버 테러 시나리오를 기술하기 위해서는 공격 트리 구조를 변형하였다. 그리고 사이버테러의 가장 대표적인 시나리오인 분산 서비스 거부 공격을 실제 실험하고 시뮬레이션 수행한 후에 결과를 비교하였다.

그러나, 네트워크 방어를 정확하게 시뮬레이션 하기 위해서는 아직 해야 할 일이 있다. 우선 다양한 사이버 침입 시나리오를 모델링한 후 시뮬레이션을 수행해 보아야 한다. 또한, 다양하고 정교한 시뮬레이션 수행을 위해 사람의 행위도 표현 가능한 동적 시뮬레이션 기능도 필요하다고 생각한다.

참 고 문 헌

[1] Smith, R and Bhattac Harya, "Firewall Placement In a Large Network Topology" in Proc. 6th IEEE workshop on Future Trends of Distributed Computing Systems, 1997.

[2 Breslau, L., et al., "Advances in Network Simulation" *Computer*, Col. 33, No.5, May 2000.

[3] Optimum Network Performance, OPNET Modeler, *http://www.opnet.com/products/modeler/home.html*, March 2001.

[4] John R. Mostow, John D. Roberts, John Bott, "Integration of an Internet Attack Simulator in an HLA Environment", Proc. IEEE workshop on Information Assurance and Security, West Point, NY, June 2000.

[5] Donald Welch, Greg Conti, Jack Marin, "A framework for an Information Warfare Simulation", Proc. IEEE workshop on Information Assurance and Security, June 2001.

[6] "SSF Simulator Implementation", http://www.ssfnet.org/ssfImplementations.html.

[7] "Domain Modeling Language(DML)", http://www.ssfnet.org/homePage.html.

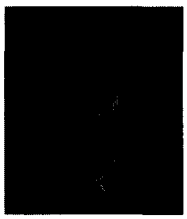
[8] Schneier, B., "Attack Tree Secrets and Lies", pp.318-333, John Wiley and Sons, New York, 2000.

[9] "Scalable Simulation Framework", http://www.ssfnet.org/homePage.html.

이 철 원(Cheol Won Lee) 정회원

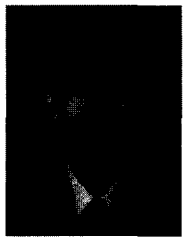
1987년 2월 : 충남대학교 수학과 졸업
1989년 8월 : 중앙대학교 전산학과 석사
2002년 2월 : 아주대학교 컴퓨터 공학과 박사수료
2000년 6월 ~ 현재 : 국가보안 기술연구소 팀장
<관심분야> 컴퓨터공학, 통신공학, 정보보호학

윤 주 범(Joo Beom Yun) 정회원



1999년 2월 : 고려대학교 컴퓨터학과 졸업
2001년 2월 : 서울대학교 컴퓨터공학과 석사
2001년 3월 ~ 현재 : 국가보안 기술연구소 연구원
<관심분야> 컴퓨터공학, 통신공학, 정보보호학

임 을 규(Eul Gyu Im) 정회원



1992년 2월 : 서울대학교 컴퓨터공학과 졸업
1994년 2월 : 서울대학교 컴퓨터공학과 석사
2002년 5월 : Univ. of Southern California, Computer Science 박사
2002년 4월 ~ 현재 : 국가보안기술 연구소 선임연구원
<관심분야> 컴퓨터공학, 통신공학, 정보보호학