情報保護學會論文誌
第18卷 第1號, 2008. 2

IARAM: 시뮬레이터를 위한 인터넷 공격 표현 및 맵핑 기법

이 철 원[1†], 김 정 식[2‡], 김 동 규[3‡]
[1]ETRI 부설 연구소, [2]한양대학교, [3]아주대학교

IARAM: Internet Attack Representation And Mapping Mechanism for a Simulator

Cheol-Won Lee[1†], Jung-Sik Kim[2‡], Dong-Kyu Kim[3]
[1]ETRI, [2]Hanyang Univ, [3]Ajou Univ.

요 약

현재 나날이 인터넷 웜의 종류 및 피해가 증가하는 상황에서 웜 공격에 대한 연구의 필요성이 증가하고 있다. 시뮬레이션은 인터넷 웜을 연구하는 데 가장 많이 사용하는 방법 중의 하나인데, 일반적으로 대규모 네트워크상에서 동작하는 웜을 시뮬레이션하기 위해서는 비용 및 시간의 문제가 발생한다. 이러한 문제를 해결하기 위해 모델링 네트워크를 사용한 시뮬레이션 방법도 많이 사용되는데, 모델링 네트워크 시뮬레이션 방법은 개별 웜을 효과적으로 적용하기 힘들다는 문제가 존재한다. 이에 본 논문에서는 대규모 인터넷 웜 시뮬레이션을 효과적으로 수행할 수 있도록 하는 웜 공격의 표현 기법과 표현된 웜 공격을 시뮬레이션 적용시킬 수 있는 매핑 기법을 제안하였다. 본 논문에서 제안하는 웜 공격의 표현 및 시뮬레이터로의 매핑 기법을 통하여 개별 웜을 좀 더 세밀히 표현할 수 있게 된다. 따라서 웜 공격 시뮬레이션 시, 모델링 네트워크 시뮬레이션의 특징인 빠른 수행 시간을 가짐과 동시에 개별 웜 공격을 세밀히 표현할 수 있게 된다.

ABSTRACT

Internet becomes more and more popular, and most companies and institutes use web services for e-business and many other purposes. With the explosion of Internet, the attack of internet worm has grown. Simulation is one of the most widely used method to study internet worms. But, it is quite challenging to simulate very large-scale worm attacks because of various reasons. By this reason, we often use the modeling network simulation technique. But, it also has problem that it difficult to apply each worm attacks to simulation. In this paper, we propose worm attack representation and mapping methods for apply worm attack to simulation. The proposed method assist to achieve the simulation efficiency. And we can express each worm attacks more detail. Consequently, the simulation of worm attacks has the time-efficiency and the minuteness.

Keywords : IARAM, 시뮬레이터, 공격 표현 기법

Ⅰ. 서 론

정보 사회가 진행되어 인터넷에 대한 의존도가 높아짐에 따라 인터넷 웜에 의한 피해도 크게 늘어나게 되었다. 인터넷 웜은 1988년 모리스 웜을 시작으로 2001년 코드 레드 웜, 2003년 슬래머 웜 등이 출현하며 막대한 피해를 입혔다. 이 중 슬래머 웜의 경우는 전 세계적

접수일: 2007년 10월 26일; 채택일: 2007년 12월 7일
† 주저자, cheolee@etri.re.kr
‡ 교신저자, bisa1004@hanmail.net

으로 최소 7만 5천대 이상의 서버가 감염되었는데, 대부분의 서버는 매우 짧은 시간 내에 감염되어 피해가 더욱 커지게 되었다[1]. 이러한 웜 공격에 대한 피해를 최소화하기 위해 다양한 웜의 특성을 연구할 필요성이 대두되었고, 시뮬레이션은 이러한 웜의 특성을 연구하는데 가장 효과적인 방법으로 알려져 왔다.

시뮬레이션이 웜을 연구하는데 효과적이지만, 일반적으로 수십만 호스트 이상으로 구성된 네트워크에서 활동하는 웜을 시뮬레이션하기 위해서는 시간과 비용면에서 큰 노력이 필요하게 된다. 그래서 이 문제를 해결하기 위해 모델링 방법을 이용한 시뮬레이션이 많이 연구되어 왔다[2]. 모델링을 이용한 시뮬레이션이란, 네트워크 또는 네트워크에서 발생하는 현상을 수학적인 식으로 이루어진 모델을 통해 시뮬레이션을 수행하는 방법이다.

이러한 방법은 네트워크의 규모가 증가하여도 수학적인 계산을 통해 네트워크의 현상을 파악하기 때문에 일반적인 시뮬레이션과 같이 수행시간이 급격히 증가하는 현상이 나타나지 않는다. 하지만 모델링을 이용한 시뮬레이션에서는 모델링된 네트워크가 실제 네트워크를 단순화, 추상화시킨 네트워크이기 때문에 시뮬레이션 수행 시 실제 패킷의 교환 등의 세부 이벤트가 표현되지 않아서 자세한 네트워크 상태를 알아보기 어렵고, 실제 네트워크에서 발생할 수 있는 현상을 동적으로 적용하기 어렵다.

다시 말하면 인터넷 웜을 시뮬레이션하기 위해서 모델링 기법을 사용하는 것이 효과적이다. 하지만 모델링 기법은 웜을 표현하기 위해서 수많은 종류의 웜을 각각 적용시켜야 한다는 문제가 발생한다. 이를 해결하기 위해서 모델링을 이용한 시뮬레이션 수행 시, 웜을 효율적으로 표현할 수 있는 방법이 필요하다.

본 논문에서는 모델링을 통한 시뮬레이션을 위해서 웜 공격을 좀 더 효과적으로 표현하기 위한 웜 공격 시나리오 표현 기법, 그리고 이렇게 표현된 시나리오를 실제 모델링된 네트워크 시뮬레이터에 매핑하는 기법을 제안한다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 네트워크 공격 표현법과 대규모 시뮬레이션에 관한 기존 연구에 대해 정리하고, 기존 방법의 문제에 대하여 정의하였다. Ⅲ장에서는 본 논문에서 제안하는 웜 공격의 표현과 시뮬레이션 방법에 대해 제안하였고, Ⅳ장에서는 실제 시뮬레이션을 위한 구조를 설계하고 Ⅴ장에서 슬래머 웜에 대한 시뮬레이션을 수행해 보았다. 마지막으로 Ⅵ장에서 향후 연구방향을 제시하고, 결론을 맺었다.

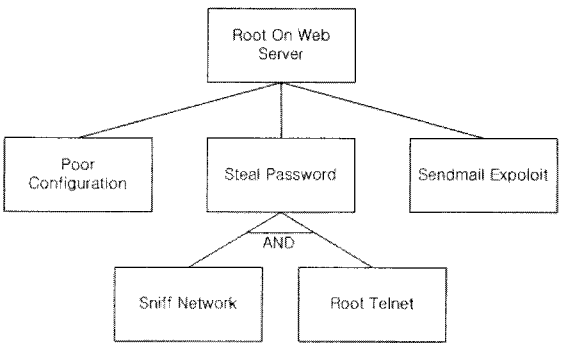
Ⅱ. 관련연구 및 문제 정의

2.1. 기존 연구

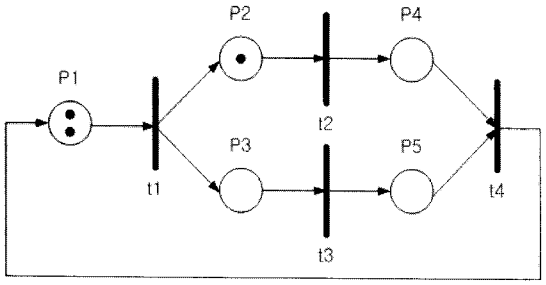
2.1.1. 공격 표현 기법

네트워크상에서 이루어지는 공격을 표현하는 방법에 대해 여러 가지 연구가 진행되었다. 트리를 사용한 방법은 그 중 하나로 공격 트리 모델(Attack Tree Model)이라고 부른다[14]. 공격 트리 모델은 루트 노드에서 시작하여 하위 노드로 진행하며 공격 시나리오를 표현해 준다. 트리의 각 노드는 공격을 성공시키기 위해 필요한 동작이고, 간선을 따라 진행하게 된다. 트리의 하위 노드들은 AND 또는 OR 연산으로 구분하게 되는데, AND 연산의 경우에는 AND로 표시된 모든 하위노드가 성공적으로 수행되어야 공격이 성공하는 것이다. OR 연산의 경우에는 하나의 하위노드만 수행되어도 공격이 성공된다.

[그림 1]은 공격 트리 모델의 예로 웹 서버의 루트 권한을 얻는 공격을 표현한 트리이다. 웹 서버의 루트 권한을 얻기 위한 방법으로는 3가지 방법이 존재하는데, 이 방법들은 OR 연산이기 때문에 이 중 1개의 동작만 성공하여도 루트 권한을 획득할 수 있다. 하지만 패스워드를 훔치는 동작을 위해서는 AND로 연결된 하위 노드의 동작이 모두 성공해야 가능하다. 이러한 공격 트리



(그림 1). 공격 트리 모델의 예

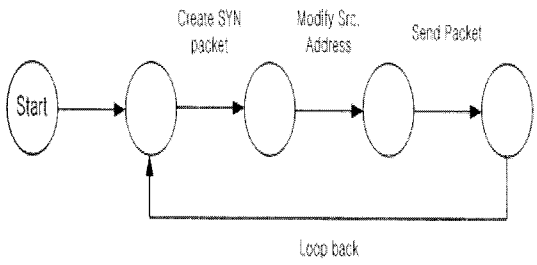


〔그림 2〕. Petri-net의 예

모델은 각 노드의 행동에 기반을 하고 있으며 공격의 각 부분을 잘 분리할 수 있지만, 트리의 구조상 동적으로 변화하는 공격을 표현하기는 힘들고 공격의 순서를 표현하기 어렵다.

네트워크 공격은 Petri-net(Place/transition net)을 사용하여 표현할 수도 있다[15]. Petri-net은 1962년 Carl Adam Petri가 개발한 표현 기법으로 place nodes, transition nodes, directed arcs를 사용하여 동작을 표현할 수 있다. [그림 2]는 Petri-net의 예로, place P1에 존재하는 토큰이 transition t1을 통과하여 P2로 이동하는 단계를 보여주고 있다. Petri-net은 현재 어떤 동작이 진행되는 지를 한눈에 알아볼 수 있다는 장점이 존재하지만, 복잡한 시나리오의 겨우, 그래프의 크기가 너무 커지기 때문에 구현이 힘들다는 문제가 있다.

STAT(State Transition Analysis Tool)은 상태전이 그래프(State Transition Diagram)를 사용하여 공격을 표현하는 방법이다[16][17]. STAT은 petri-net과 같이 토큰이 존재하지 않으며, state와 transition 두 가지로만 구성된다. STAT에서는 공격 시나리오가 시작되면 각 동작과 transition이 일치하는 지를 판단하게 된다. 만약 두 동작이 일치하게 되면 해당 transition이 수행되며 시나리오가 진행되게 된다. 동작이 일치하지 않을 경우에



〔그림 3〕. STAT의 예

는 새로운 공격 시나리오를 그래프에 추가할 수 있다. [그림 3]은 STAT의 간단한 DoS 공격을 표현한 예이다.

공격이 시작되면 패킷 생성, 주소 변경, 패킷 전송의 과정이 반복되며 진행되게 된다. STAT의 단점은 transition과 해당 동작이 정확히 일치하지 않으면 다른 공격의 동작으로 판단한다는 점이다. 이 경우 새로운 공격 시나리오로 간주되어 그래프에 추가되기 때문에 유사 공격의 구분이 힘들게 된다.

Bordeleau 등은 계층 구조를 가진 그래프를 사용한 표현 방법을 제안하였다[18]. 이 방법은 scenario interaction, scenario dependency, scenario clustering 이란 개념을 사용하여 시나리오의 동작간의 관계를 표현할 수 있게 해주었다. Lee 등은 이와 비슷하게 계층적 상태 전이 그래프(Hierarchical State Transition Graph, HSTG)를 사용하여 네트워크의 공격을 탐지하는 방법을 제안하였다[19]. 계층 구조를 사용하게 되면 동작간의 관계를 표현할 수 있기 때문에, STAT의 단점인 동작의 유사성을 효과적으로 표현할 수 있게 되었다.

위와 같이 네트워크 공격을 표현하기 위한 방법에는 여러 가지가 존재하는데, [표 1]은 방법들 간의 차이를 보여주고 있다. 본 논문에서는 표에서 보듯이 단위 공격 간의 관계를 효율적으로 표현하여 수많은 웜 공격들을 분류해 줄 수 있는 HSTG를 기반한 방법을 사용하게 된다.

2.1.2. 모델링 네트워크 시뮬레이션

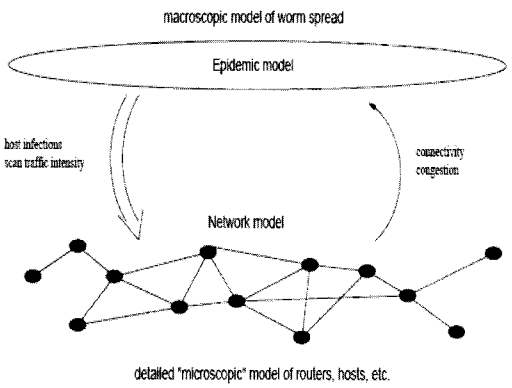
대규모 네트워크를 시뮬레이션 하기 위해서는 모델링 기법을 사용하는 것이 효과적이다. 모델링 기법은 여러 종류가 있는데, 대규모 네트워크 시뮬레이션에는 유

〔표 1〕. 공격 표현 기법의 비교

	Tree-based	Petri-net	STAT	HSTG
시나리오 표현	O	O	O	O
다중 시나리오 표현	X	O	O	O
단위 공격간의 관계 표현	X	X	X	O
단위 공격의 그룹화	O	X	X	O

체 모델, Epidemic 모델 등을 이용한 모델링 방법이 많이 사용되고 있다. 유체 모델을 이용한 방법에는 Misra 등이 TCP 트래픽 모델링을 위해 제안한 동적 유체 흐름 모델(Dynamic Fluid Flow Model)이 있다[24]. 그리고 Liu 등은 이 모델링 방법을 수정한 모델링 방법을 제안하였는데 이로 인해 네트워크 내부의 트래픽 전달을 좀 더 정확하게 표현할 수 있게 되었다[3]. Liu 등이 제안한 수정된 유체 모델을 살펴보면, 트래픽 흐름에 대하여 동일한 경로와 전송 지연 값을 갖는 클래스들을 정의하고, 각 노드에서의 출발 비율과 도착 비율을 계산하여 전체 네트워크의 흐름을 표현하였다. 이러한 유체 모델을 이용한 모델링 방법의 주요 목적은 백본 네트워크에 흐르는 대규모 트래픽을 효과적으로 표현하기 위한 것이다. 하지만 이러한 모델링 방법을 이용하여 패킷레벨의 트래픽 변화를 효과적으로 표현하는 데는 한계가 있다. Liu 등에 의한 수정된 유체 모델링 방법에서는 패킷 트래픽의 흐름을 미리 정의하고 이러한 패킷 트래픽과 대규모 백본 트래픽을 함께 고려하여 모델링에서 사용되는 인자 값을 구한 후에 이를 이용하여 모델링을 수행하였다. 이러한 수정된 방법에서도 패킷 트래픽의 변화에 따른 전체 트래픽의 변화는 표현하기 어려운 실정이다.

Epidemic 모델을 이용한 모델링 방법은 원래 생물학 분야에서 생물학적 바이러스 등의 감염을 모델링할 때 많이 사용되어온 모델링 방법이었다. 이 모델링 방법에서는 감염가능 객체 수, 감염된 객체 수 및 치료된 객체 수를 계산해 주는 모델로, 웜의 전파를 연구하는데 효과



[그림 4]. Epidemic 모델을 사용한 시뮬레이션 시 네트워크 구성도

적인 모델이다[4][5]. 이 모델에서는 전체 호스트의 개수를 N개라고 가정할 때, 다음 수식을 통해 네트워크의 상태를 파악하게 된다.

$$\frac{ds(t)}{dt} = -\beta s(t)i(t)$$
$$\frac{di(t)}{dt} = \beta s(t)i(t) - \gamma i(t)$$
$$\frac{dr(t)}{dt} = \gamma i(t) \qquad (1)$$
$$\text{단, } s(t) + i(t) + r(t) = N, \forall t \geq 0$$

수식 (1)에서 상수 β는 감염 파라미터 (감염 비율), γ는 삭제 파라미터, s(t)는 t 시점에서의 감염가능(susceptible) 호스트 수를 나타내고, i(t)는 t 시점에서의 감염된(infected) 호스트 수를, r(t)는 t 시점에서의 치료된(removed) 호스트 수를 나타낸다. 이 모델은 네트워크 호스트의 수가 충분히 커서 확률적 시스템의 변화는 충분히 예측할 수 있고, 호스트 사이의 상호작용이 거의 일정하다고 가정하였다.

Epidemic 모델을 이용하여 시뮬레이션을 수행할 경우, [그림 4]와 같이 네트워크를 광범위 모델(Macroscopic model)과 극소범위 모델(Microscopic model)로 나누어 시뮬레이션을 수행하게 된다. 광범위 모델은 호스트의 감염을 나타내고, 극소범위 모델은 백본 네트워크의 연결을 표현한다. 즉, 백본 네트워크의 연결여부 및 속도에 따라서 어느 서브네트워크가 어느 정도의 속도로 감염되는지가 결정되게 된다. 하지만 이 시뮬레이션에서는 광범위 모델에 중점을 두었기 때문에 극소범위 모델의 정보교환은 대부분 시뮬레이션을 수행하기 전에 인자값이 결정되는 정적인 정보의 교환이 이루어지게 된다.

이외에도 패킷 단위 네트워크와 모델링 네트워크가 혼재되어 동작하는 여러 기법이 제안되었는데, Global Mobile Information System Simulator (GloMoSim)[10]은 유체 모델을 이용하는 컴포넌트와 패킷 단위의 시뮬레이션을 수행하는 모델로 나뉘어져 있으며, 유체 모델을 통하여 전송지연과 패킷 손실을 계산하여 트래픽의 흐름을 표현하고, 패킷 단위의 모델에서는 유체 모델의 트래픽을 고려하여 개별 패킷의 전송 지연과 손실율을 계산하게 된다. 이 모델링 방법에서는 유체 모델에서 표현되는 트래픽 량이 패킷 단위의 모델을 이용하여 표현되는 트래픽 량보다 월등히 많으므로, 두 모델사이의 상호작용은 무시된다. 이와 유사한 연구로서

Hybrid Discrete-Continuous Flow Network Simulator (HDCF-NS)[11]와 Kiddle[12] 등이 제안한 혼성 기법이 있다. 또한, Kalyan S. Perumalla[9] 등은 실제 네트워크와 가상 네트워크가 함께 동작하는 high-fidelity 모델링 방법을 제안하였다. 이 기법들은 실제 네트워크를 구성했다는 점에서 일부 측면에서 시뮬레이션의 유연성을 크게 향상시켰다고 할 수 있으나, 실제 네트워크와 가상 네트워크 사이의 트래픽 교환 표현 방법, 실제 네트워크의 확장성 부족 등의 한계로 인하여 이러한 방법의 활용이 제한되고 있다. 하지만 파라미터의 업데이트를 통하여 패킷 단위 네트워크와 모델링 네트워크를 연동한 hybrid 모델에 대한 연구도 진행되어, hybrid 모델을 사용하여 웜을 시뮬레이션 하여 실제 상황과 근접한 결과를 보여주기도 하였다[2].

2.2. 문제 정의

모델링된 네트워크에서 웜 시뮬레이션을 효과적으로 수행하기 위해서는 시뮬레이션하려고 하는 웜을 세밀히 표현할 수 있는 기법이 필요하다. 모델링 네트워크의 시뮬레이션은 웜의 고유한 성질을 적용시켜야 하는데, 웜의 표현이 잘못된다면 시뮬레이션의 결과가 달라지기 때문이다. 하지만 웜은 각기 다양한 특성을 가지고 있으며, 짧은 기간에 여러 변종이 출현하는 경우가 다수 존재한다. 그렇기 때문에 다양한 웜을 신속하고 효과적으로 시뮬레이션하려고 할 때, 웜을 효과적으로 표현하고 이를 모델링된 네트워크에 매핑하여 시뮬레이션을 수행할 수 있도록 하는 방법이 필요하다.

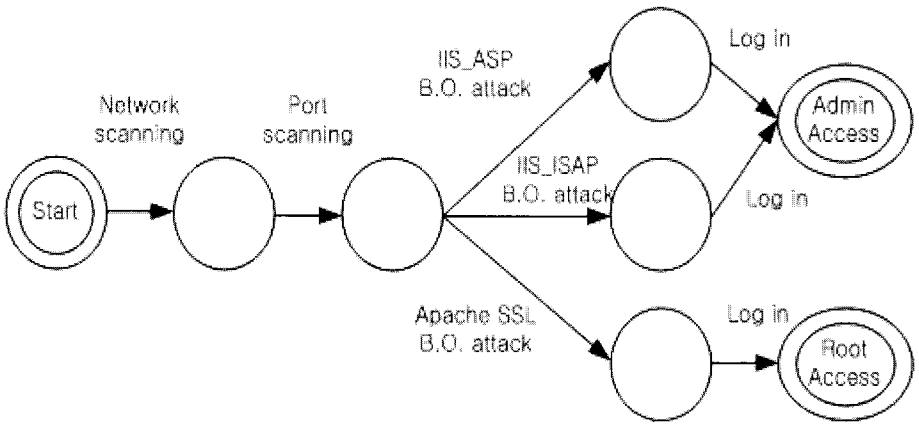
기존의 웜 시뮬레이션 방법은 모델링된 네트워크에 개별 웜을 직접 프로그래밍하여 시뮬레이션을 수행하기 때문에 웜 공격을 시뮬레이션하기 위해서는 전문 프로그래머의 노력을 필요로 한다. 또한 변종과 같이 약간의 변화를 적용하기 위해서도 프로그래밍 된 코드를 일일이 수정해 주어야 한다. 따라서 웜 공격을 효율적으로 표현하고, 이를 모듈화 하여 웜을 직접 프로그래밍하지 않고 시뮬레이션을 할 수 있는 방법이 필요하다. 또한 이렇게 작성된 모듈을 사용하여 웜을 모델링 네트워크에 적용하기 위한 웜 시나리오 매핑 기법이 필요하다.

Ⅲ. 제안하는 방법

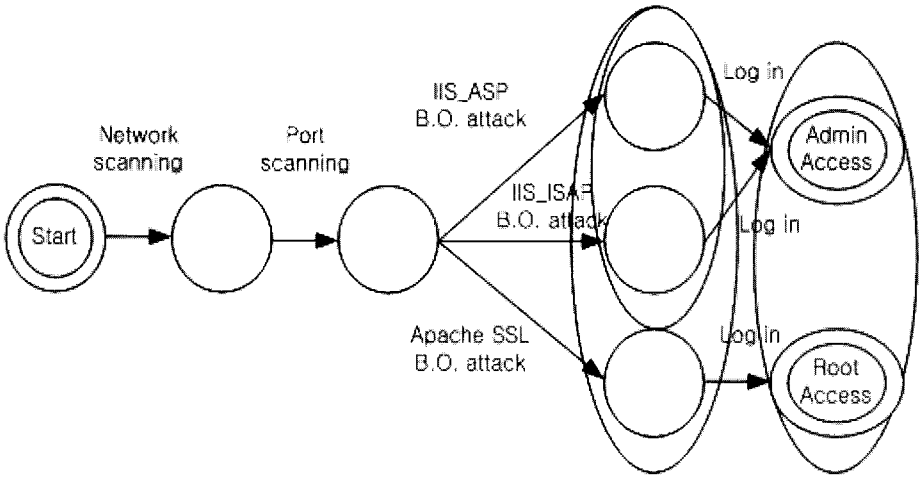
본 논문에서는 효율적인 웜 시뮬레이션을 위해서는 웜 공격을 표현한 웜 시나리오 표현기법과 이렇게 표현된 웜 공격을 시뮬레이션 할 수 있도록 시뮬레이션 환경에 매핑하는 기법을 제안한다. 본 논문에서는 Lee 등이 제안한 HSTG[19]에 기초한 웜 표현 기법을 제안하고, 이를 시뮬레이션 환경에 매핑하는 기법을 제안한다.

3.1. 웜 공격 표현 기법

웜 공격을 시나리오 형태로 표현하는 방법에는 공격 트리 모델을 이용하는 방법[14], Petri-net을 이용하는 방법[15], 상태전이 그래프를 이용하는 방법[16][17][18][19] 등이 있는데 본 논문에서는 HSTG를 이용한 방법을 기초로 하여 웜 공격 시나리오를 표현할 수 있도록 하였다. HSTG를 이용한 방법은 공격의 진행 상태를 노드로 표



〔그림 5〕. 상태 전이 그래프를 이용한 공격 표현의 예



[그림 6]. HSTG를 이용한 공격 표현의 예

현하고, 단위 공격 또는 행위를 간선을 이용하여 나타낸다. 또한 여러 개의 공격 시나리오를 하나의 그래프로 표현해 줄 수 있다. 다중 시나리오를 표현할 수 있게 된다면 다음과 같은 장점이 존재한다.

- 상호 연관성 파악용이
- 공통의 단위 공격 파악 및 공용
- 탐지 규칙 자동 생성용이

[그림 5]는 상태 전이 그래프를 사용하여 버퍼 오버플로우 공격을 표현한 예이다. 그림에서는 3 가지의 유사한 공격 시나리오를 보여주고 있다. 각 시나리오의 순서는 공격이 시작된 이후, 네트워크 검색, 포트 검색, 버퍼 오버플로우 공격의 순서로 진행되어 관리자 또는 루트 권한을 획득하는 것이다. 좀 더 상세히 말하면, 각 버퍼 오버플로우 공격은 CERT Coordination Center에서 발표한 IIS ASP 버퍼 오버플로우 공격(CAN-2002-0079), IIS ISAPI 버퍼 오버플로우 공격(CAN-2001-0241), Apache mod_SSL 버퍼 오버플로우 공격(CAN-2002-0082)을 나타낸다.

[그림 5]의 문제는 3 가지의 공격 시나리오가 연관성 없이 표현되어 있다는 점이다. 이와 같이 간단한 시나리오가 아닌 많은 수의 시나리오가 표현되어야 하는 경우에는, 유사성 등의 관계가 전혀 표현되어 있지 않기 때문에 각 시나리오를 별개로 취급해 주어야한다. 그렇기 때문에 HSTG에서는 연관성의 표현을 위해서 유사한 상태들에 대하여 이를 대표할 수 있는 상위 상태를 생성하고, 이렇게 생성된 상위 상태에 유사한 상태들을 묶어 하위 상태로 추가한다. 예를 들어, 일반적인 버퍼 오버플로우 공격이라는 상위의 상태를 생성하여, 이 상위 상태에Apache SSL 버퍼 오버플로우 공격과 IIS 웹 서버 버퍼 오버플로우 공격을 하위의 상태로 추가한다. [그림 6]은 [그림 5]에 계층 구조 및 연관성을 적용시킨 HSTG 공격 표현법의 예이다.

본 논문에서는 여러 개의 웜 공격 시나리오를 하나의 그래프 상에서 표현할 수 있는 HSTG를 사용하여 웜 공격을 표현한다.

여러 개의 웜 공격 시나리오를 하나의 그래프 상에서 표현하기 위해서는 이미 존재하는 그래프 G에 새로운 시나리오 S1을 추가하는 방법이 있어야 한다. 새로운 시나리오 S1을 추가하기 위해서는 S1에 있는 각 상태와 그래프 G상에 있는 모든 상태를 비교해 보아야 한다. 이러한 비교 연산은 S1의 상태 수나 그래프 G의 상태수가 증가함에 따라 기하급수적으로 증가하게 된다. 이러한 문제는 해결하기 위하여 HSTG와 마찬가지로 웜 공격 시나리오를 계층적인 표현방법을 이용하여 그래프 상에 표현하였다. 이러한 계층적인 표현을 위하여 다음과 같은 상태간의 관계가 그래프 상에 표현될 수 있어야 한다.

- 유사 관계 (Similarity) : 인터넷 웜은 하나의 웜인 경우에도 많은 변종이 존재하고, 서로 다른 웜인

경우에도 일부 유사한 상태가 존재하게 된다. 이렇게 서로 다른 시나리오에서 비슷한 상태들이 존재하게 될 때 이들 사이의 공통요소를 추출하여 효과적으로 이러한 관계를 표현할 수 있는 메커니즘이 필요하다.

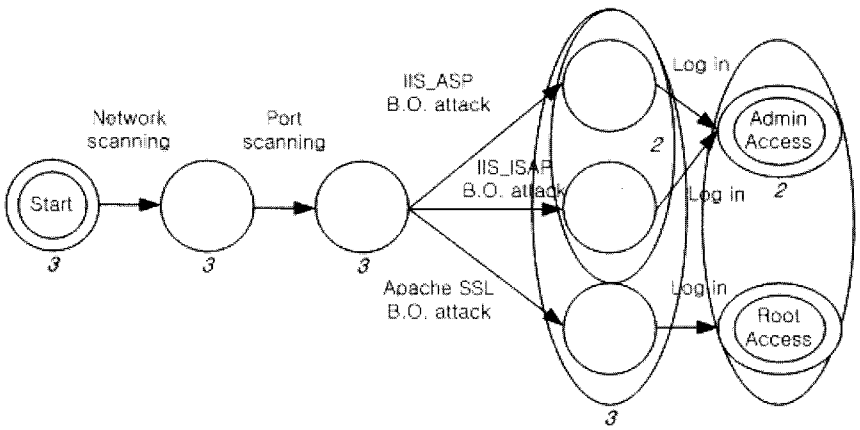
- 포함 관계 (Inclusion) : 만약 상태 A가 상태 B의 특수한 경우라면, 상태 A와 상태 B는 포함 관계가 있다고 할 수 있다. 즉, 상태 B가 상태 A를 포함한다.
- 전이 관계 (Transition) : 웜 공격 시나리오는 상태 전이 그래프 상에 표현될 때, 상태들 간의 전이를 표현함으로써 표현된다. 따라서 이러한 상태들 간의 전이를 간선(edge)으로 표현할 수 있다.

HSTG는 위와 같은 관계들을 효과적으로 표현할 수 있도록 설계되었다. HSTG에서는 상태들 사이의 공통요소를 추출하고, 이러한 공통 성분을 그래프 상에 반영하기 위하여 계층적 구조를 만든다. HSTG는 여러 계층의 상태를 가지며, 상태를 계층적으로 표현함으로써 상태의 복잡도를 감소시키고 상태전이 수를 감소시킨다. 따라서 새로운 시나리오 추가 시, 상태간의 비교연산 횟수를 크게 감소시킬 수 있다.

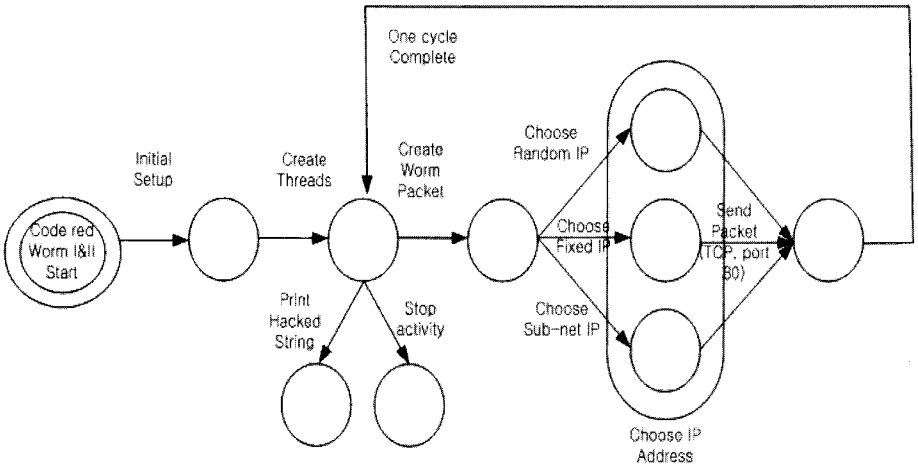
HSTG에서는 하나의 상태가 하나 이상의 부 상태(sub state)를 가질 수 있고, 하나의 상태는 다른 상태로 계층에 상관없이 상태 전이를 할 수 있다. HSTG를 이용하여 인터넷 공격 시나리오를 표현할 때, 새로운 취약점이 발견되고 이를 이용한 새로운 공격 시나리오가 가능하다면 새로운 상태와 이에 따른 상태 전이가 정의되고 이렇게 새롭게 정의된 상태와 상태전이는 쉽게 기존의 그래프에 합병될 수 있다. HSTG의 장점 중의 하나는 새로운 시나리오가 그래프에 추가될 때 발생하는 비교 연산 횟수를 계층적인 구조를 사용함으로써 감소시킬 수 있다는 것이다. 또한 상태들 사이의 유사 관계, 포함 관계 등이 그래프 상에 표현됨으로써 그래프에 표현된 웜들 사이의 관계가 보다 명확해 지고, 이를 이용한 여러 가지 변종 웜 등의 발견이 가능해 진다.

본 논문에서는 HSTG의 표현법에 기초하여 여러 웜 공격을 더 효율적으로 표현할 수 있도록 가중치를 사용한다. 가중치는 각 상태에 부여되게 되는데, 현 상태가 몇 개의 시나리오에서 사용되고 있는지를 알려주는 역할을 한다. 일반적으로 같은 계열의 웜 공격은 거의 유사한 동작이 다수 존재하는데, 가중치를 사용하면 현재 상태에 관여한 시나리오의 개수를 판단하여 그래프의 표현력을 더욱 높여주게 된다. 가중치는 각 상태에 부여되며, 하위 상태와는 별도로 상위 상태에도 각각 부여해주게 된다. [그림 7]은 가중치를 부여한 HSTG의 예이다. 그림은 3가지의 시나리오를 포함하고 있기 때문에, 각 상태는 1에서 3까지의 가중치를 가지고 있게 된다. 단, 가중치가 1인 상태는 숫자를 표시하지 않았다.

가중치를 부여하게 되면, 몇 가지 연산을 통해 웜 공격에 대한 그래프의 생성을 더욱 쉽게 할 수 있다.



[그림 7]. 가중치가 부여된 HSTG의 예



〔그림 8〕. Code-red worm I & II의 그래프

일반적인 웜 공격 시나리오는 다음의 각 연산의 조합을 통해 그래프에 추가, 수정, 삭제 및 관계의 표현을 할 수 있게 된다.

- Insert : 그래프에 동작을 추가시켜 주는 연산이다. 새로운 웜 공격을 표현해 주고자 할 경우, 하나의 동작단위로 추가를 해주게 된다. 삽입 연산은 추가하려는 상태를 현재 그래프에 존재하는 상태들과 비교하여 유사, 중복인 상태를 파악하게 된다. 상태가 유사할 경우에는 subset 연산을 통하여 그룹을 생성하게 되고, 중복인 상태인 경우에는 가중치를 증가시켜주게 된다.
- Delete : 삭제 연산은 그래프의 축소 및 최적화 과정에서 이루어지게 된다. 삭제는 하나의 상태를 삭제하게 되는데, 다른 공격 시나리오에 영향을 미치지 않으면서 이루어져야 한다. 그래서 각 시나리오 별로 부여되어 있는 가중치의 감소도 담당하게 된다.
- Split : 분할연산은 상태를 더욱 자세하게 표현하고자 할 경우나 새로운 상태가 삽입될 경우, 이전 상태에서 다음상태로 넘어가는 간선이 1개 이상으로 분할해주는 연산이다. 이 연산은 복수의 공격 시나리오가 그래프에 표현될 경우, 각 시나리오가 진행하는 방향을 결정하게 된다.
- Subset : 계층적 표현방법에서 중요한 역할을 하는 그룹을 생성하는 역할을 한다. 그룹은 하위 상태를 포함한 상위 상태를 생성하게 되며, 새로 생성된 상위 상태는 포함된 모든 하위 상태의 가중치의 합을 자신의 가중치로 가지게 된다.

3.2. 공격 매핑 기법

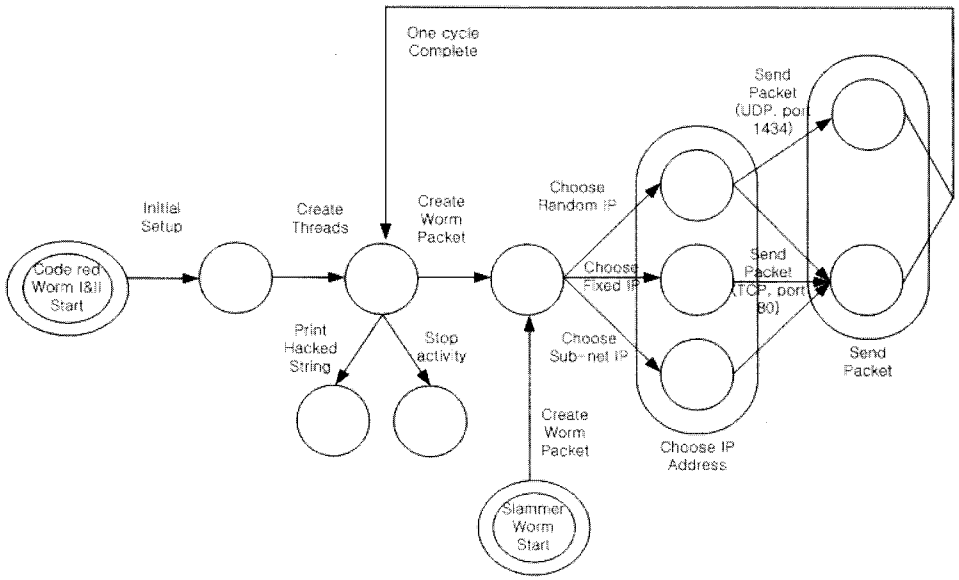
HSTG는 웜 공격 시나리오를 효과적으로 표현해 줄 수 있지만, 실제 시뮬레이션을 수행하기 위해서는 그래프와 시뮬레이션 환경을 매핑 시켜주는 과정이 필요하다. 본 논문에서는 다음의 순서로 진행되는 매핑 방법을 제안한다.

1) 상태 전이 그래프 작성
2) 그래프의 각 간선(Edge)에 대한 매핑 모듈(Mapping module) 생성/탐색
3) 파라미터 설정 및 시뮬레이션 수행

3.2.1. 상태 전이 그래프 작성

웜 공격 시나리오의 매핑을 위해서는 시뮬레이션을 수행해야하는 웜의 공격 단계에 맞추어 계층적 상태 전이 그래프로 작성을 하여야 한다.

예를 들어 RCS(Random Constant Spreading) 형태의 웜[20]인 슬래머 웜(Slammer Worm)[23]에 대한 그래프를 작성해 보았다. 이 경우, 새로운 그래프가 생성되는 것이 아니라, 기존에 작성되어 있는 그래프에 슬래



〔그림 9〕 Slammer Worm의 추가 그래프

머 웜을 추가하는 경우를 생각하였다. [그림 8]은 기존에 작성되어 있는 그래프로, 슬래머 웜과 유사한 특성을 지는 코드 레드 웜[21][22]을 표현한 그래프이다. 이 그래프는 기존 코드레드 웜과 변종인 코드레드 웜 II를 같이 표현하였다. 여기에 슬래머 웜의 동작과정을 추가한 그래프가 [그림 9]이다. 슬래머 웜은 패킷 생성, 랜덤 IP 주소 선택, 패킷 전송의 단계를 반복해서 동작하게 된다. 이는 코드 레드 웜보다 간단한 동작과정이고, 두 웜이 비슷한 단계가 많기 때문에 패킷 생성, 랜덤 IP 주소 선택의 상태는 기존 그래프의 상태를 그대로 사용하게 된다. 그리고 패킷을 보내는 과정은 TCP와 UDP라는 차이점이 존재하지만 유사한 연산이기 때문에 같은 그룹을 지어 연관성을 표시해 주게 된다.

3.2.2. 매핑 모듈 생성/탐색

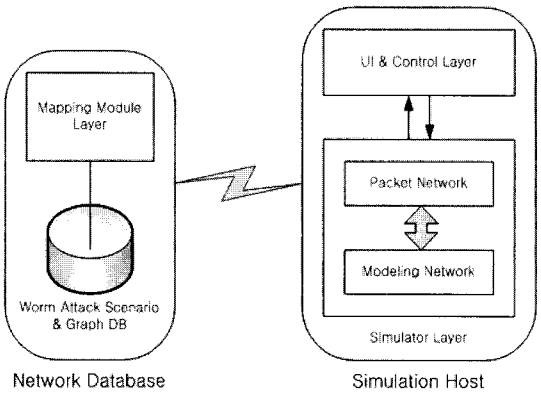
상태 전이 그래프로 표현된 웜 공격 시나리오는 각 동작이 간선으로 표현되어 있다. 웜 시뮬레이션에 공격 시나리오를 신속하고 효과적으로 적용하기 위해서는 각 간선을 시뮬레이션에 자동으로 적용시켜 줄 수 있는 매핑 모듈이 존재하여야 한다. 매핑 모듈의 작성은 시뮬레이터의 종류에 맞추어 작성을 하여야 한다. 예를 들어 event-driven 방식의 네트워크 시뮬레이터인 SSFNet[6]의 경우에는 DML이라는 환경설정 파일을 사용하여 시뮬레이션을 진행하게 된다. SSFNet에서 사용되는 매핑 모듈의 경우에는 이 DML 파일을 자동으로 수정해 줄 수 있도록 하는 것이 편리하다. 그리고 매핑 모듈은 동작의 유사성, 포함관계 등을 판단하여 기존에 작성된 모듈을 재사용하여 작성을 할 수 있기 때문에, 공격 시나리오의 양이 많아지게 될수록 더욱 효과적으로 매핑 모듈을 작성할 수 있게 된다.

[그림 9]의 경우에는 패킷의 생성, IP 주소 선택, 패킷 전송의 동작을 Create_Packet( ), ChooseIP( ), SendPacket( )과 같은 매핑 모듈로 작성하여 시뮬레이터에 적용할 수 있다. 또한 ChooseIP( ) 모듈을 상속받아 그 하위 관계에 속한 ChooseRandomIP( ), ChooseFixedIP( )와 같은 모듈을 작성하게 되는 것이다.

3.2.3. 파라미터 설정 및 수행

매핑 모듈을 작성하면 실제 웜 공격에 대한 시뮬레이션 설정을 시작하게 된다. 웜의 동작에 대한 매핑 모듈은 작성되어 있기 때문에, 이를 시뮬레이터의 환경에 적용이 될 수 있도록 모듈을 수행시켜 준다. 모듈의 수행은 기존에 작성되어 있는 공격 시나리오에 기초하여, 모듈을 순서대로 수행시켜 준다.

SSFNet의 경우에는 매핑 모듈로 수정된 DML 파일에 시뮬레이션에 필요한 환경설정 값을 작성하여 환경



〔그림 10〕. 시뮬레이션 구조

설정을 마무리한 뒤, 시뮬레이션을 시작하게 된다.

Ⅳ. 시뮬레이션 구조

본 논문에서 제안한 웜 공격 표현 기법 및 매핑 기법은 웜 공격을 효과적으로 표현하고, 이를 시뮬레이터에 적용시키는 기법을 소개하고 있다. 본 논문에서는 이 기법을 사용하여 시뮬레이션을 수행하는 시스템 구조를 설계해 보았다. 시스템은 네트워크 시뮬레이터인 SSFNet을 사용하여 수행한다는 것을 가정하여 구조를 설명하였다.

시뮬레이션 구조는 크게 네트워크 데이터베이스와 시뮬레이션을 수행하는 호스트로 나누어진다. 네트워크 데이터베이스는 웜 공격 시나리오와 이를 표현한 그래프, 매핑 모듈을 저장하는 중앙 저장소의 역할을 수행하고, 시뮬레이션 호스트는 실제 사용자가 시뮬레이션을 제어할 수 있도록 사용자 인터페이스와 컨트롤 계층과 시뮬레이터 계층으로 구성되어 실제 시뮬레이션을 수행해 준다. [그림 10]은 전체 시스템의 구조를 보여 준다.

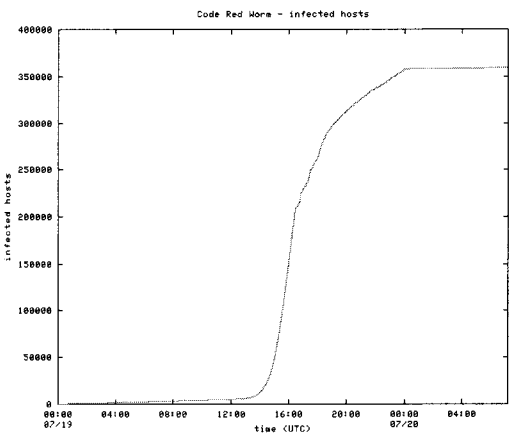
데이터베이스를 네트워크로 별개로 저장하는 이유는 웜 공격의 방대함 때문이다. 웜 공격을 표현하고 그 동작의 매핑 모듈을 생성하는 작업은 전문 인력의 노력이 필요한 작업이다. 각기 다른 시뮬레이션을 위해 공통적으로 사용할 수 있는 모듈 등을 각자의 공간에서 작성하게 되면, 효율성 높이기 위한 취지와 맞지 않게 된다. 그렇기 때문에 수많은 웜들의 공격 시나리오 및 매핑 모듈을 네트워크를 통하여 하나의 데이터베이스에 저장하면 중복되는 작업을 많은 부분 줄일 수 있게 된다.

실제 시뮬레이션을 수행하는 위치인 시뮬레이션 호스트는 네트워크 데이터베이스와 통신을 하고 사용자가 제어하는 시뮬레이션 환경 설정을 담당하는 사용자 인터페이스 & 제어 계층이 존재한다. 이 계층에서 시뮬레이터의 환경 설정 파일인 DML의 설정을 제어해 주게 된다. 또한 시뮬레이션이 수행되는 가상 네트워크는 웜 공격을 표현하는 모델링 네트워크와 모델링 네트워크를 동적으로 업데이트 해주는 역할을 하는 패킷 네트워크로 구성하게 된다. 패킷 네트워크를 사용하여 동적 파라미터를 업데이트하게 되면, 웜 공격의 동작을 더 상세하게 모델링 네트워크에 적용할 수 있게 된다[2].

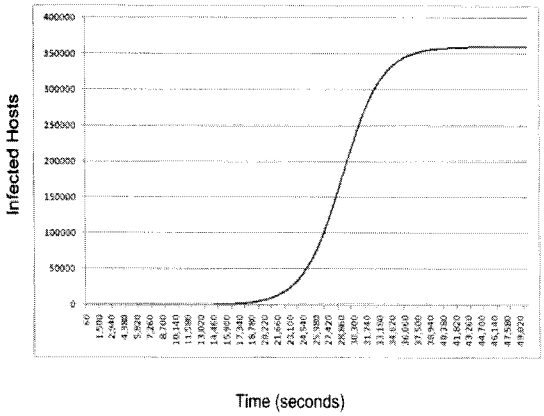
Ⅴ. 실 험

웜 공격을 그래프로 표현하고 이를 자동으로 매핑해 주는 방법은 시뮬레이션 시간을 크게 단축시킬 수 있게 된다. 본 논문에서는 제안하는 방법을 사용하여 패킷 네트워크와 모델링 네트워크를 연동시킨 혼성 네트워크[2]에서 간단한 실험을 진행하였다. 실험은 코드레드 웜에 대한 시뮬레이션 환경을 그래프의 매핑기법을 통해 슬래머 웜에 대한 시뮬레이션 환경으로 변환하여 시뮬레이션을 수행하도록 한다. 시뮬레이션에 코드레드 웜과 슬래머 웜을 선택한 이유는 두 웜 모두 RCS 형태의 웜으로, 동작 간에 공통점이 상당히 존재한다는 점이다. 시뮬레이션은 RedHat Linux Enterprise ver.3에서 수행되었으며, 시뮬레이터는 SSFNet ver. 2.0을 사용하였다.

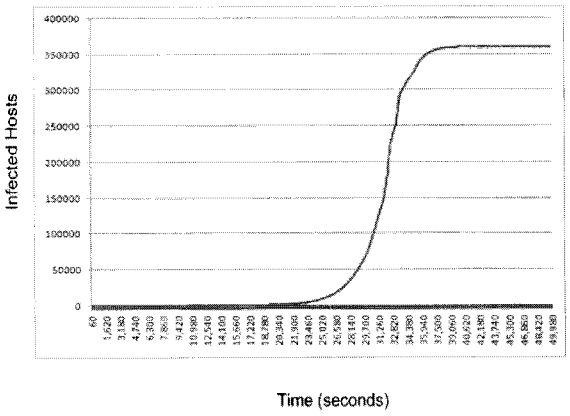
실험은 코드레드 웜을 먼저 시뮬레이션 하였다.



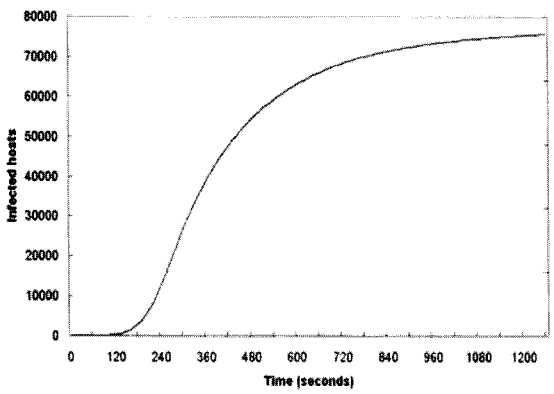
〔그림 11〕. 실제 코드레드 웜 전파 그래프



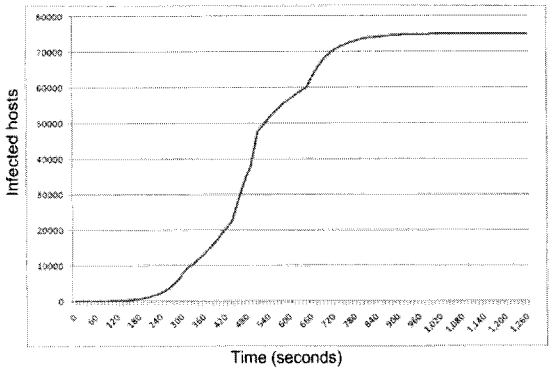
〔그림 12〕. 기존 연구의 코드레드 웜 전파 시뮬레이션 결과 그래프



〔그림 13〕. 혼성 네트워크에서의 코드레드 웜 전파 시뮬레이션 결과 그래프



〔그림 14〕. 기존 연구의 슬래머 웜 전파 시뮬레이션 결과 그래프



〔그림 15〕. 혼성 네트워크에서의 슬래머 웜 전파 시뮬레이션 결과 그래프

[그림 13]은 코드레드 웜을 시뮬레이션 한 결과로 패킷 네트워크를 통해 동적인 현상까지 적용이 된 시뮬레이션 결과이다. 그래프의 결과가 코드레드 웜을 잘 표현했는지를 확인해 보기 위해, 실제 코드레드 웜의 전파 그래프와 이를 시뮬레이션 한 기존 연구의 그래프 [그림 11], [그림 12]로 추가하였다[5][6][8]. 기존 연구의 시뮬레이션 결과 그래프를 추가한 이유는 실제 코드레드 웜의 전파 그래프와 시뮬레이션 결과 그래프의 x축 단위가 틀리기 때문에, 혼성 네트워크의 시뮬레이션 결과 그래프와 같은 단위를 가진 결과로 비교를 하기 위함이다.

이렇게 작성된 코드레드 웜의 시뮬레이션 환경을 본 논문에서 제안한 방법을 사용하여 슬래머 웜에 대한 시뮬레이션 환경으로 변경해 주게 된다. 시나리오 매핑에 사용될 그래프는 [그림 9]를 사용하게 된다. 그림에서는 슬래머 웜은 패킷의 전송 방법이 UDP로 코드레드 웜의 TCP와 다르다는 것을 알 수 있게 된다. 이 차이에 의해서 슬래머 웜의 전파 속도가 훨씬 빨라지게 되는 것이다. 실험에서는 전송 방법에 대한 차이를 자동으로 변경해 주기 위해 Slammer_SendPacket( )이라는 모듈을 작성하였다. [그림 9]의 시나리오 그래프에서 시뮬레이션에 적용되는 부분은 전송 방법에 따른 전파 속도의 차이이기 때문에, Slammer_SendPacket( )의 모듈을 실행시킨 후, 시뮬레이션을 수행해 보았다.

[그림 15]는 시뮬레이션의 결과로 매핑 모듈을 통해 코드레드 웜에서 슬래머 웜으로 자동으로 변경된 환경을 사용하였다. 슬래머 웜도 코드레드 웜과 마찬가지로 비교를 위한 기존 시뮬레이션 결과를 [그림 14]로 추가하였다[25]. 여기서 본 논문에서 시뮬레이션을 수행한 결과의 정확도는 기존의 결과와 비교를 통하여 확인할 수 있다.

〔표 2〕. 시뮬레이션 환경 설정 방법의 비교

	사용자 수동 설정	공격 매핑 기법 사용
시뮬레이터 설정 (초기 1회)	혼성 모델 모듈 작성 환경 설정 파일(DML) 수정	혼성 모델 모듈 작성 매핑 모듈 작성 매핑 모듈 실행
웜 전환 시 (매 전환 시)	혼성 모델 모듈 수정 환경 설정 파일(DML) 수정	매핑 모듈 실행

[표 2]는 매핑 모듈을 사용할 경우와 그렇지 않을 경우에 필요한 작업을 비교한 것이다. 매핑 모듈을 사용하지 않을 경우에는 혼성 모델을 준비하는 과정은 동일하지만 매핑 모듈을 작성할 필요가 없기 때문에, 시뮬레이터 설정 단계에서는 매핑 모듈을 사용하는 경우에 비해서 더 빠른 시간에 작업을 종료할 수 있게 된다. 하지만 공격 매핑 기법을 사용하게 되면 매핑 모듈은 작성을 한 뒤에는 이후 웜을 전환하여 실험을 할 경우에는 매핑 모듈을 실행하기만 하면 되지만, 매핑 모듈을 사용하지 않을 때에는 웜을 전환할 때 마다 혼성 모델의 모듈을 현재의 웜에 맞추어 수정을 해 주어야 한다. 또한 사용자가 환경 설정 파일을 수정하는 시간에 비해서 매핑 모듈을 실행시키는 시간이 상대적으로 적게 걸리게 된다.

본 장에서는 실험은 2가지의 웜만을 시뮬레이션 하였지만, 제안한 방법을 확장하여 사용할 경우 다양한 웜 공격에 대한 시뮬레이션을 효율적으로 수행할 수 있게 된다.

Ⅵ. 결 론

웜 공격을 연구하는데 많이 사용하는 대규모 네트워크 시뮬레이션 방법은 시간, 비용의 문제를 상당부분 해결해주게 된다. 하지만, 수많이 발생하는 웜 공격에 대해 일일이 시뮬레이터를 설정하여 시뮬레이션을 수행하는 것은 여전히 많은 노력이 필요한 상황이다. 이를 해결하기 위해 본 논문에서는 HSTG에 기초하여 웜 공격을 표현하고, 이를 시뮬레이션 환경에 매핑해 주는 매핑 모듈에 대한 기법을 제안하였다.

본 논문에서 제안하는 방법은 웜 공격을 계층적 상태 전이 그래프를 사용하여 웜을 표현해 주게 된다. 계층적 상태 전이 그래프는 유사한 웜, 변종을 연관시키고, 그룹을 만들어 웜의 동작을 단계적으로 표현해 주게 된다. 이렇게 표현된 그래프는 매핑 모듈을 통해 빠르게 시뮬레이터에 적용되게 된다. 또한, 공간의 문제와 노력의 절감을 위해 네트워크 데이터베이스와 시뮬레이션 호스트로 구성된 시뮬레이션 시스템 구조를 설계하였다.

웜 공격을 효과적으로 시뮬레이션 할 수 있는 방법에는 여러 가지가 있겠지만, 본 논문에서는 공격 표현 기법과 그 활용을 사용한 방법을 제시함으로, 웜 공격을 연구할 수 있는 방법에 대해 알아보았다. 그리고 웜 공격 그래프 크기의 감소, 매핑 모듈의 자동화의 연구를 진행하여 더욱 빠르고 정확하게 웜을 시뮬레이션 할 수 있도록 연구할 계획이다.

참고문헌

[1] David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford, Nicholas Weaver. "The Spread of the Sapphire/Slammer Worm". *CAIDA Technical report*, 2003.

[2] 김정식, 박진호, 조재익, 최경호, 임을규, "RCS 웜 시뮬레이션을 위한 Hybrid 모델링 방법", *한국정보보호학회논문지(1598-3986)*, 제17권 3호, pp. 43-53, 2007년 6월

[3] Yong Liu, Francesco Lo Presti, Vishal Misra. "Fluid Models and Solutions for Large-Scale IP Networks". *ACM SIGMETRICS Performance Evaluation Review*, Volume 31, Issue 1, pp. 91-101, 2003.

[4] D.J.Daley, J.Gani. "*Epidemic modelling : an introduction*", Cambridge University Press, 1999

[5] Michael Liljenstam, David M. Nicol, Vincent H. Berk, Robert S. Gray. "Simulating Realistic Network Worm Traffic for Worm Warning System Design and Testing". *The ACM WORM 2003*, pp.24-33.

[6] SSFNet Web page. http://www.ssfnet.org/

[7] NS-2 Web page. http://www.isi.edu/nsnam/ns/

[8] David Moore, Colleen Shannon. "The Spread of the Code-Red Worm (CRvw)". *CAIDA Analysis page*. www.caida.org/analysis/security /code-red/coderedv2_analysis.xml

[9] Kalyan S. Perumalla, Srikanth Sundaragopalan. "High-Fidelity Modeling of Computer Network Worms". *The 20th Annual Computer Security Applications Conference(ACSAC'04)*, pp. 126-135, 2004.

[10] Xiang Zeng, Rajive Bagrodia, Mario Gerla. "GloMoSim: A library for parallel simulation of large-scale wireless networks". *The 12th Workshop on Parallel and Distributed Simulation*, pp. 154-161, 1998.

[11] Benjamin Melamed, Shuo Pan, Yorai Wardi. "Hybrid discrete-continuous fluid-flow simulation". *SPIE*, Volume 4526, pp. 263-270, 2001.

[12] Cameron Kiddle, Rob Simmonds, Carey Williamson, Brian Unger. "Hybrid packet/fluid flow network simulation". *The Seventeenth Workshop on Parallel and Distributed Simulation (PADS'03)*, pp. 143, 2003.

[13] Stuart Staniford, Vern Paxson, Nicholas Weaver. "How to Own the Internet in Your Spare Time". *The 11th USENIX Security Symposium*, pp. 149-167, 2002.

[14] T. Tidwell, R. Larson, K.Fitch, J.Hale, "Modeling Internet Attacks", T*he 2001 IEEE Workshop on Information Assurance and Security*, pp. 54-59, 2001.

[15] Richard Zurawski, MengChu Zhou, "Petri Nets and Industrial Applications: A Tutorial", *The IEEE Transactions on Industrial Electronics*, VOL. 41, No. 6, 1994.

[16] K. Ilgun, R. A. Kemmerer, P. A. Porras, "State Transition Analysis: A Rule-Based Intrusion Detection System", *The IEEE Transactions on Software Engineering*, VOL. 21, No. 3, pp. 181-199, 1995.

[17] Giovanni Vigna, Steve T. Eckmann, Richard A. Kemmerer, "The STAT Tool Suite", *The IEEE DARPA Information Survivability Conference and Exposition 2000*, Vol. 2, pp. 46-55, 2000.

[18] F. Bordeleau, J. P. Corriveau, B. Selic, "A scenario-based approach to hierarchical state machine design", *The Third IEEE Symposium on Object-Oriented Real-Time Distributed Computing*, pp. 78-85, 2000.

[19] Lee C.W., Im E.G., Chang B.H., Kim D.K., "Hierarchical state transition graph for internet attack scenarios", *The International Conference on Information Networking 2003*, 2003.

[20] Stuart Staniford, Vern Paxson, Nicholas Weaver. "How to Own the Internet in Your Spare Time". T*he 11th USENIX Security Symposium*, pp. 149-167, 2002.

[21] Technical Report, "ANALYSIS: .ida "Code Red" Worm", *eEye Digital Security*, http://research.eeye.com:80/html/advisories/published/AL20010717.html, 2001.

[22] Technical Report, "ANALYSIS: CodeRed II Worm", http://research.eeye.com/html/advisories/published/AL20010804.html, *eEys Digital Security*, 2001.

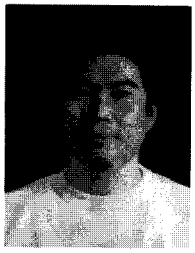
[23] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, N. Weaver, "The Spread of the Sapphire/Slammer Worm", *CAIDA Technical report*, http://www.caida.org/publications/papers /2003/sapphire/, 2003.

[24] Vishal Misra, Wei-Bo Gong, Don Towsley, "Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED", *ACM SIGCOMM Computer Communication Review*, Volume 30,

Issue 4, pp. 151-160, 2000.

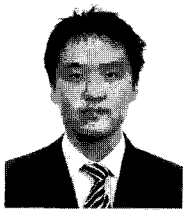
[25] Songjie Wei, Jelena Mirkovic, Martin Swany, "Distributed Worm Simulation with a Realistic Internet Model", *The Workshop on Principles of Advanced and Distributed Simulation (PADS'05)*, pp. 71-79, 2005.

〈著者紹介〉



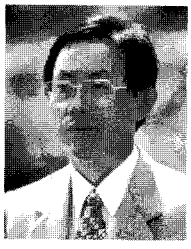
이 철 원 (Cheol-Won Lee) 정회원

1987년 : 충남대학교 수학과 졸업
1989년 : 중앙대학교 대학원 전자계산학과 석사
2001년 : 아주대학교 대학원 컴퓨터공학과 박사과정 수료
1989년~1996년 : 한국전자통신연구원 선임연구원
1996년~2000년 : 한국정보보호센터 선임연구원/과제 책임자
2000년 ~ 현재 : ETRI부설연구소 책임연구원/실장
2003년 ~ 2004년 : Texas A&M 대학교 방문연구원
<관심분야> 컴퓨터 및 네트워크 보안, 정보통신기반보호, 정보보호시스템 평가기준



김 정 식 (Jung-Sik Kim) 학생회원

2006년 2월 : 한양대학교 컴퓨터전공 학사
2006년 3월~현재 : 한양대학교 전자컴퓨터통신공학 석사
<관심분야> 센서 네트워크, 시뮬레이션, 네트워크 보안



김 동 규 (Dong-Kyu Kim) 정회원

1973년 2월 : 서울대학교 공과 대학 응용수학과 졸업
1979년 2월 : 서울대학교 자연과학대학원 전자계산학과 석사
1984년 : 미국 Kansas State University 전자계산학과 박사
1986년 : IEEE 802.4, 802.6, 802.10 Working Group Member
1979년~현재 : 아주대학교 정보 및 컴퓨터 공학부 교수, Asiacrypt '96 조직위원장, 건설교통부 항공 교통관제소 신공항 교통관제 시스템 평가위원회 위원, 한국과학기술연구소 연구원, 한국통신학회 상임이사, 한국정보보호학회 부회장 역임
<관심분야> 컴퓨터 통신, 정보보호, 프로토콜 엔지니어링