인터넷 보안 시뮬레이션을 위한 공격 모델링

서 정 국[†] · 최 경 희[††] · 정 기 현[†††] · 박 승 규[††] · 심 재 홍[††††]

요 약

최근 인터넷 사용이 폭발적으로 증가함에 따라 인터넷은 다양한 공격에 노출되었다. 이러한 인터넷 공격을 시뮬레이션 하기 위해서는 공격을 효과적으로 모델링 할 수 있어야 한다. 그러나 기존의 인터넷 공격 모델링 기법들은 공격을 단순히 특징에 따라 분류하거나 종류를 나누는 데 중점을 두고 있으며, 인터넷 보안 시뮬레이션을 위한 공격 시나리오를 표현하는데 있어서는 적합하지 않았다. 본 논문에서는 기존의 트리 기반 공격 모델링 기법을 보완하여 인터넷 보안 시뮬레이션의 공격 모델링 기법으로 활용할 수 있게 개선하였다. 개선된 공격 모델링 기법은 복잡한 시나리오의 표현 불가, 공격 실행 순서의 모호함, 시스템 상태 정보의 결여 등과 같은 기존 트리 기반 모델링 기법의 문제점들을 해결하였다. 또한 기존 모델링 기법으로는 기술할 수 없었던 동시간 공격 표현, 정밀한 공격 시작 및 수행기간 지정 등이 가능하도록 하였다.

Attack Modeling for an Internet Security Simulation

Jung-kuk Seo[†] · Kyung-hee Choi[††] · Gi-hyun Jung[†††]
Seung-kyu Park[††] · Jae-hong Sim[††††]

ABSTRACT

As the use of the Internet has explosively increased, it is likely for the Internet to be exposed to various attacks. Modeling the Internet attacks is essential to simulate the attacks. However, the existing studies on attack modeling have mainly focused on classifying and categorizing the attacks and consequently they are not suitable to representing attack scenarios in the Internet security simulation. In this paper, we introduce the existing methods of attack modeling, and propose an adapted attack modeling to properly express the properties for the Internet security simulator. The adapted attack modeling suggests a solution to the problems of the existing attack tree modelings, such as difficulty of composing complex scenarios, ambiguity of attack sequence, lack of system state information. And it can represent simultaneous, precise time-dependent attack, and attack period, which are nearly impossible to be represented in many other existing methods.

키워드 : 정보전(Information Warfare), 인터넷 공격(Internet Attack), 인터넷 보안 시뮬레이션(Internet Security Simulation), 공격 모델링(Attack Modeling)

1. 서 론

현대 사회는 고도로 정보화되어 일반 가정에서부터 국가 기간 시설에 이르기까지 컴퓨터와 인터넷이 폭 넓게 사용되고 있다. 이처럼 사회 전 분야에 걸쳐서 정보 통신에 대한 의존도가 커짐에 따라 이에 대한 보안이 중요한 문제가 되었다. 초기의 인터넷 공격은 단순히 해커의 실력 과시용으로 사용되어 그 위험성이 적었지만, 지금은 각종 범죄에 이용되거나 더 나아가 국가적 차원에서 정보전에까지 이용되기에 이르렀다.

※ 이 논문은 국가지정연구실(National Research Laboratory) 사업의 지원에 의해 연구되었음.
† 준 회 원 : 아주대학교 대학원 정보통신학과
†† 정 회 원 : 아주대학교 정보통신학과 교수
††† 정 회 원 : 아주대학교 전자공학부 교수
†††† 정 회 원 : 조선대학교 인터넷소프트웨어공학부 교수
논문접수 : 2003년 12월 31일, 심사완료 : 2004년 2월 16일

인터넷 공격에 대비하기 위해 다양한 인터넷 공격과 이들이 네트워크에 미치는 영향에 대한 연구가 필수적이다. 그러나 인터넷 공격에 대한 실험을 실제 네트워크에서 수행하기에는 경제적, 시간적으로 많은 제약이 따르고, 실제 실험용 공격이 이루질 경우 네트워크에 심각한 손상을 입힐 가능성도 크다. 이런 점에서 가상 환경의 시뮬레이션을 통한 인터넷 공격 연구는 언급된 문제점을 극복하는 훌륭한 연구 방법이 될 수 있다[1, 2]. 인터넷 공격을 가상 환경에서 시뮬레이션 하기 위해서는 무엇보다도 인터넷 공격들을 적절히 표현하고 공격간 연관성을 잘 표현하여 시뮬레이션 시나리오로 구성할 수 있도록 해주는 공격 모델링 기법이 반드시 필요하다.

최근의 인터넷 공격은 매우 복잡한 형태로 진행이 되고 있다. 분산 서비스 거부 공격(DDoS)이나 인터넷 웜 공격(Internet worm attack)에서 볼 수 있듯이 분산된 많은 공

격이 동시 다발적으로 진행되고, 또한 공격자는 자신의 목적을 달성하기 위해 다양하고 연계된 공격들을 다단계로 구성하여 목표 대상을 타격하기도 한다[14]. 그러나 기존의 인터넷 공격에 대한 모델링 기법은 주로 취약점이나 공격 목적에 따라 종류를 나누어 분류 하거나[3-5], 또는 분류된 공격 종류별로 통계학적 빈도에 따른 중요성을 제시하는 것이 대부분이었다[6]. 이런 공격 분류의 대표적인 예는 미국 MITRE 법인에서 발표하는 CVE(Common Vulnerabilities and Exposures)[7]와 NIST(National Institute of Standards and Technology)의 ICAT[8] 연구이다. 이 연구들은 인터넷 공격의 공통 이름 영역(namespace)을 정의하고 각 공격들의 발생 빈도에 따른 중요도를 제공한다. 이런 분류의 방법을 통한 연구는 유용한 인터넷 공격에 대한 통찰을 제공하기는 하지만 공격 상호간의 연관성에 대한 정보는 알려 주지 못하므로, 인터넷 보안 시뮬레이션을 위한 공격 모델링에는 적합하지 않다. 그런데 최근에 와서 공격 상호간의 연관성을 기술하고자 하는 노력이 이루어지고 있다. 사용 목적에 따라 다양한 모델링 기법이 있으나 기반으로 하고 있는 표현 방법에 따라 크게 그래프 기반 공격 모델링(attack graph modeling)[9-11] 기법과 트리 기반 공격 모델링(attack tree modeling)[12-14] 기법으로 나눌 수 있다.

그러나 이들 모델링 기법은 시스템 보안 테스트, 특정 네트워크의 취약점 및 침입 가능성 테스트, 침입 탐지 시스템의 탐지 룰(rule) 기술 등의 목적으로 개발되었으므로 인터넷 보안 시뮬레이션을 위한 공격 모델링 기법으로 사용하기에는 적합하지 않았다.

본 논문에서는 기존 두 모델링 기법을 기반으로 인터넷 공격 시뮬레이션에 적합한 개선된 공격 모델링 기법을 제안한다. 개선된 공격 모델링 기법은 트리 기반 공격 모델링을 기반으로 하되 트리 기반 공격 모델링의 문제점을 보완하고 그래프 기반 모델링의 복잡한 시나리오의 표현의 장점을 반영하였다. 따라서 복잡한 시나리오의 표현 불가, 공격 실행 순서의 모호함, 시스템 상태 정보의 미 반영 등과 같은 기존 문제점들을 해결하였다. 또한 기존 모델링 기법으로는 기술할 수 없었던 동시간 공격 표현, 정밀한 공격 시작 및 수행기간 지정, 구현 편의성 제공 등이 가능하도록 하였다.

본 논문의 구성은 다음과 같다. 2장에서 그래프 기반 공격 모델링 기법과 트리 기반 공격 모델링 기법에 대해 자세히 살펴보고, 이들의 장단점을 분석한다. 3장에서는 기존 공격 모델링 기법을 기반으로 인터넷 보안 시뮬레이션에 적합한 개선된 공격 모델링 기법을 제시한다. 4장에서는 개선된 모델링 기법을 기반으로 공격 시나리오 예제를 모델링해 봄으로써 개선된 모델링 기법의 타당성을 검증한다. 마지막 5장에서 결론을 맺는다.

2. 관련 연구

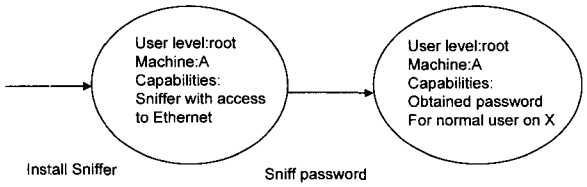
그래프 기반 공격 모델링과 트리 기반 공격 모델링 기법은 공격들 상호간의 연관성을 기술하기 위한 목적으로 개발 되었으며, 새로이 개발된 시스템들의 보안 테스트, 특정 네트워크의 취약점 및 침입 가능성 테스트, 침입 탐지 시스템의 탐지 룰(rule) 기술 등 다양한 분야에서 활용되고 있다. 본 절에서는 이 두 모델링 기법의 개념과 장단점에 대해 살펴보고, 인터넷 보안 시뮬레이션을 위한 인터넷 공격 시나리오 기술(description)에 이들을 활용하는데 있어 어떠한 한계성을 가지는지 살펴본다.

2.1 그래프 기반 공격 모델링

그래프 기반 공격 모델링[9-11] 기법은 노드와 노드를 연결하는 선으로 이루어진 그래프를 이용해 공격 상호간의 연관성을 기술한다. 기본적으로 그래프 기반 공격 모델링 기법은 설정 파일(configuration file), 공격자 개요(attacker profile), 공격 템플릿(attack template)으로 구성된다. 설정 파일은 네트워크 토폴로지를 표현하고, 각 네트워크 노드의 설정 상태, 즉 운영체제나 동작하고 있는 프로그램 등에 대한 설정 정보를 기록한다. 공격자 개요는 공격자가 가진 기술 정도, 공격 프로그램 등을 나타내고, 이 정보는 공격 그래프의 생성시에 각 공격의 성공 여부를 판단하는 기준으로 사용된다. 공격 템플릿은 잘 알려진 공격 방법에 대해 미리 작은 공격 그래프의 형태로 구성해 놓은 것으로, 이 템플릿을 바탕으로 다른 공격과의 연계된 공격을 공격 그래프로 구성할 수 있도록 해 준다.

그래프의 노드는 공격의 상태를 표현하고, 노드들을 연결하는 선은 공격 프로그램의 실행, 공격자에 의해 가해진 행위, 공격의 상태 변화 등을 나타낸다. 노드는 사용자(user level : 사용자의 권한 정도, 예를 들어 루트 계정, 일반 계정 등), 시스템(machine : 공격과 연관이 있는 컴퓨터 혹은 네트워크 구성 요소), 취약성(vulnerabilities : 공격자에 의해 야기된 원래 설정의 변화), 능력(capabilities : 네트워크에 대한 물리적 접근이나 프로그램을 실행할 수 있는 권한) 등과 같은 구성요소로 이루어진다.

(그림 1)은 암호 훔쳐보기 공격을 표현하는 공격 템플릿이다. 이 공격은 루트 계정을 가진 컴퓨터(A)에 패킷을 훔쳐보는 프로그램(sniffer)을 설치하여 같은 랜 영역 사용자의 패킷을 훔쳐보는 공격으로서, 주로 다른 컴퓨터(X)에 접속하기 위한 암호를 훔치기 위해 사용된다. 이 공격 템플릿에는 프로그램의 설치와 암호를 훔치는 행위가 노드를 연결하는 선으로 표현되고, 공격 프로그램이 설치된 상태와 암호를 탈취한 상태가 노드로 표현되어 있다.



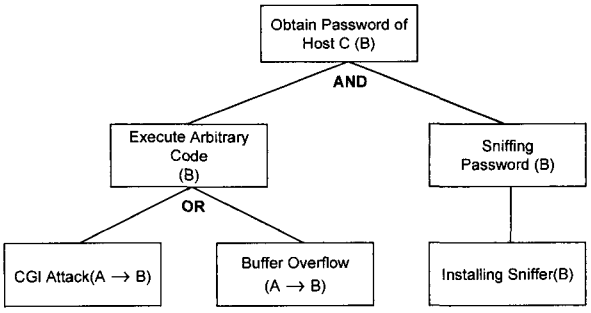
(그림 1) 암호 훔쳐보기 공격 템플릿

(그림 2)는 (그림 1)에서 설명된 암호 훔쳐보기 공격 템플릿을 포함하는 완성된 형태의 공격 그래프를 나타낸다. 이 공격 그래프에는 암호 훔쳐보기 공격 템플릿 이외에도 익명 사용자 FTP 공격(anonymous FTP attack)과 암호 추측 공격(password guessing attack)을 함께 사용하여 공격자의 목표 컴퓨터의 계정과 암호를 훔치는 과정과 그 과정에서 각 공격들간의 상호 연관성을 표현하고 있다. 또한 이 공격 그래프는 공격자의 초기 권한과 암호 훔쳐보기 공격의 결과에 따라 총 세 가지의 공격 시나리오를 포함하고 있다.

2.2 트리 기반 공격 모델링

트리 기반 공격 모델링[12-14] 기법은 노드와 노드들의 계층적 트리 구조를 공격 모델링에 사용하며, 달성하고자 하는 공격 목표를 상위 노드로 두고 목표를 이루기 위해 수행되어야 하는 공격을 하위 노드로 구성한다. 각 하위 노드는 다시 자신을 목표로 가지는 또 다른 하위 노드를 가지는 계층적 구조로 표현될 수 있다. 각 노드는 하위 노드들에 대해 AND와 OR의 논리적 연산 특성을 가진다. AND 연산으로 표현된 노드는 자신의 모든 하위 노드의 공격이 성공하였을 경우에만 자신의 공격이 수행될 수 있고, OR 연산자로 표현된 노드는 자신의 하위 노드 공격 중 하나만 성공하더라도 공격이 수행될 수 있다.

이 모델링 기법 역시 그래프 기반 공격 모델링 기법과 유사하게 설정 파일, 공격자 개요, 공격 템플릿을 가지지만 공격 템플릿과 전체 공격 시나리오가 트리 형태로 표현된다는 점이 다르다. 또한 트리의 노드가 해당 공격을 기술하고, 노드를 연결하는 선은 노드 사이의 연관성 AND와 OR를 이용해 표현한다는 점이 다르다.

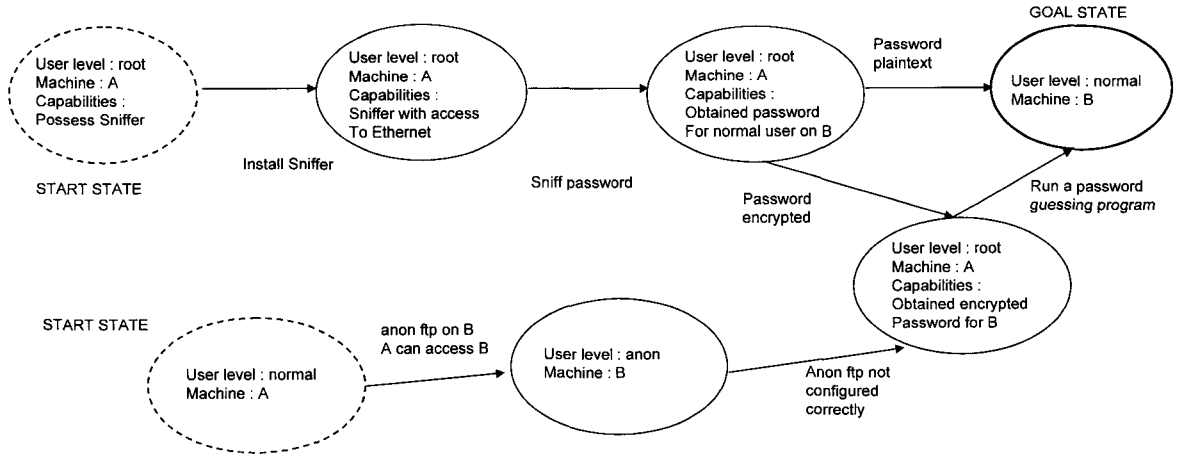


(그림 3) 트리 기반 공격 모델링

(그림 3)은 트리 기반 공격 모델링 기법을 이용한 예로, 앞서 그래프 기반 공격 모델링 기법에서 사용된 암호 훔쳐보기 공격(sniffing password)을 포함하여 트리 기반 공격 모델링 기법으로 표현한 것이다. 그림에 보인 공격 트리는 공격자 컴퓨터 A에서 컴퓨터 B의 취약점을 이용해 루트 권한을 획득한 다음 컴퓨터 B에서 스니핑 공격을 이용해 최종적으로 컴퓨터 C의 계정을 획득하기 위한 공격 시나리오이다. 이 시나리오는 CGI 공격이 성공하는 경우와 버퍼 오버 플로우 공격이 성공하는 경우의 두 가지 공격 시나리오를 가진다.

2.3 공격 모델링 기법 분석

<표 1>은 앞서 언급한 두 모델링 기법을 비교 분석한 것이다. 그래프 기반 모델링 기법의 경우 공격 시나리오가 (그림 2)의 예제보다 훨씬 더 커진다면 그래프의 복잡성으



(그림 2) 그래프 기반 공격 모델링

로 인해 시나리오를 이해 하기가 어렵다. 트리 기반 모델링 기법 역시 시나리오가 커짐에 따라 가독성은 점점 떨어지지만 계층적으로 구성된 특성에 의해 그래프 기반 모델링 기법에 비해서는 다소나마 우수하다고 할 수 있다.

〈표 1〉 모델링 기법 분석 결과

모델링 기법 구분 / 특성	그래프 기반 모델링 기법	트리 기반 모델링 기법
사용자 가독성	낮 음	높 음
재사용성	높 음	높 음
복잡한 시나리오의 표현	가 능	어려움
수행순서 (다중 공격 경로 존재 시)	미 지정	미 지정
시스템 상태 정보	반 영	미 반영
시뮬레이션 구현 편의성	어려움	쉬 움
동시간 공격 표현	불 가	불 가
수행 시작 및 기간 지정	불 가	불 가

복잡한 공격 시나리오의 표현 가능성 측면에서는 그래프 기반 모델링 기법이 우수하다. (그림 2)의 암호 훔쳐보기 공격의 경우 암호화된 결과와 평문 형태 결과 등 두 가지 형태의 결과를 얻을 수 있다. 이런 경우 동일한 공격을 트리 기반 모델링 기법으로 표현한 (그림 3)에서 알 수 있듯이 트리 구조의 특성상 공격의 성공 결과가 두 가지 이상되면 표현하기가 어려워진다.

두 모델링 기법 모두 공격 템플릿을 미리 구성해 놓고 공격 시나리오 구성 시 기존의 공격 템플릿을 재활용함으로써 재사용성을 지원할 수 있다. 공격의 수행 순서 측면에서는 그래프 기반 모델링 기법의 경우 노드를 연결하는 선의 화살표 방향으로 표현하고, 트리 기반 모델링 기법은 하위 노드에서 상위 노드로 왼쪽 노드에서 오른쪽 노드로 수행하는 것을 원칙으로 한다. 그러나 두 모델링 기법은 모두 여러 가지 공격 경로가 있을 경우 실행순서를 지정할 수 없다. 트리 기반 공격 모델링 기법은 공격 노드 자체의 연관성에 중점을 두고 시나리오를 구성하기 때문에, 공격이 성공하기 위해 필요한 프로그램의 설정 상태와 같은 시스템 상태 정보들(그래프 기반 모델링 기법의 경우 노드에서 기술됨)은 특별히 기술되지 않는다.

그러나 기존의 두 모델링 기법은 시스템들의 보안 테스트, 특정 네트워크의 취약점 및 침입 가능성 테스트, 침입 탐지 시스템의 탐지 룰 기술 등의 목적으로 개발되었다. 따라서 이들을 인터넷 보안 시뮬레이션을 위한 공격 모델링 기법으로 사용하려고 할 때는 다음과 같은 사항들이 추가로 고려되어야 한다. 인터넷 보안 시뮬레이터에서 각각의 공격은 독립된 단위의 프로그램으로 구현되어야 한다. 따라서 시뮬레이션 구현 편의성 측면에서 하나의 노드로 각각의 프로그램으로 구현된 공격을 표현할 수 있는 트리 기반 모델링 기법이 공격 실행 시나리오를 구성하는데 있어서 보다 편리하다. 그러나 기존의 두 공격 모델링 기법은 최근에 자주 사용되는 분산 서비스 거부 공격이나 웜 공격과 같이 두 가지 이상의 공격이 동시에 수행되는 공격 시나리오의 표현이 불가능하다. 또한 각각의 공격에 대해 정밀한 실행 시간 지정이 불가능하므로 특정 시간에 동작을 하도록 설정된 웜이나 바이러스 같은 경우에는 표현이 어렵다. 또한 두 명 이상의 공격자가 서로 연계하여 공격하는 경우도 역시 두 모델링 기법 모두 표현할 수 없다. 뿐만 아니라, 공격의 수행 순서에 있어서 동시에 수행되지 않는 두 가지 공격 경로가 존재하는 경우, 시뮬레이션에서는 이들 공격들에 대한 시뮬레이션 시작과 끝이 모호해 진다. 따라서 명확한 공격 시작시간과 수행 기간의 지정이 필요하다.

3. 개선된 트리 기반 공격 모델링

본 절에서는 기존 공격 모델링 기법의 분석 결과를 참고하여 인터넷 보안 시뮬레이션에 적합한 개선된 공격 모델링 기법을 제시한다. 개선된 모델은 사용자 가독성을 높이기 위해 트리 기반 공격 모델링 기법을 기반으로 하되, 앞 절의 분석에서 지적된 트리 기반 공격 모델링 기법의 문제점을 보완한다.

3.1 공격 시나리오 모델링 기법의 확장

3.1.1 기능에 따른 노드의 세분화

트리 기반 공격 모델링 기법은 모든 노드를 공격자의 행위로 정의하고, 하위 노드는 상위 노드 공격을 실행하기 위해 사전에 수행되어야 하는 공격으로 정의하였다. 개선된 모델에서는 노드를 공격자의 행위를 표현하는 **행위자 노드**(*actor node*)와 행위자 노드들을 묶고 다른 상위 노드들과 연결하는 기능을 가진 **추상 노드**(*abstract node*)로 구분한다. 노드 사이의 연산을 표현한 AND와 OR는 추상 노드에서만 지정할 수 있다. 따라서 행위자 노드는 하위 노드를 가지지 않으므로 트리의 종단에만 존재할 수 있다. 추상 노드의 이름은 하위 행위자 노드의 공격 특성을 반영하여 작명할 수 있다.

추상 노드는 그 노드가 가지는 AND와 OR 연산과 동시 실행 여부를 결합하여 **순차 AND**(*sequential AND*), **순차 OR**(*sequential OR*), **동시 AND**(*concurrent AND*), **동시 OR**(*concurrent OR*) 등의 네 가지 노드로 세분된다. 순차 AND 추상 노드는 자신의 하위 노드를 지정된 순서로 실행하고, 모든 하위 노드의 실행이 성공적으로 끝난 경우에 자신도 성공한 것으로 판단한다. 순차 OR 노드는 자신의

하위 노드를 지정된 순서로 실행하되, 하위 노드 중 하나라도 성공한 경우에 자신도 성공한 것으로 판단한다. 동시 AND 노드는 자신의 하위 노드를 동시에 수행하고, 모든 하위 노드가 성공하였을 경우 자신도 성공한 것으로 판단한다. 동시 OR 노드는 자신의 하위 노드를 동시에 수행하고, 하위 노드 중 하나라도 성공한 경우 자신도 성공한 것으로 판단한다.

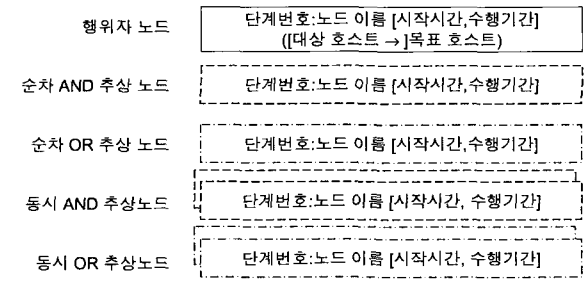
이러한 추상 노드의 확장은 행위자 노드들 사이의 관계를 보다 유연하게 표현할 수 있어, 복잡한 시나리오 구성이 가능하고 동시 AND와 동시 OR 노드를 통해 여러 개의 행위자 노드가 동일한 시간대에 수행되는 시나리오도 표현 가능하다.

3.1.2 공격 순서에 따른 단계 번호 부여

기존의 트리 기반 공격 모델링 기법은 공격의 최종 목표, 즉 마지막으로 수행되는 노드를 루트 노드로 구성하고 그 루트 노드를 수행하기 위해 선행되어야 하는 공격들을 하위 노드로 구성하였기 때문에 여러 개의 공격 경로가 존재하는 것이 가능했다. 시뮬레이션의 관점에서 볼 때 이러한 구조는 처음 시작 노드가 모호해지고 공격의 실행 흐름을 명확하게 표현하기가 어려운 문제점이 있다. 따라서 시뮬레이션에서는 실행 순서에 따라 트리를 구성하는 것이 더 적합하다. 이를 위해 먼저 트리 기반 모델링 기법의 모든 노드에게 계층적 레벨과 실행 순서에 따라 단계 번호(step number)를 부여한다. 트리의 각 노드에 단계 번호를 부여하여 표현하면 노드의 트리 구조상의 위치를 확인하기 쉽고 또한 시뮬레이션 실행 순서가 명확해지는 이점이 있다.

3.1.3 공격의 시작시간 및 수행기간 지정

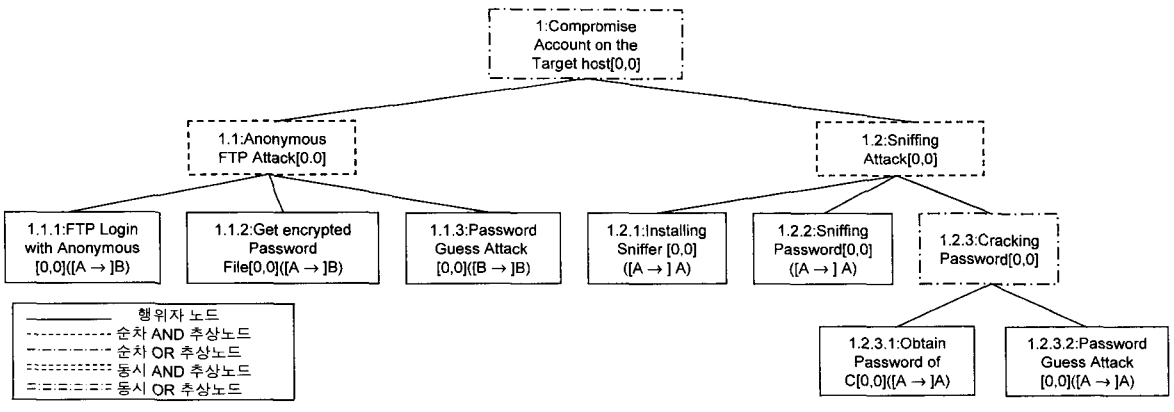
각각의 노드에 공격 노드가 실행되어야 하는 명확한 **시작시간**과 **수행기간**을 공격자가 지정할 수 있도록 한다. 시간의 지정은 시뮬레이션이 시작된 이후 경과하는 절대시간을 기준으로 표기할 경우에는 [절대시간, 수행기간]으로 표기하고, 노드가 실행 가능하게 된 시점에서 바로 시작할 경우에는 0으로 표기하고, 지정 시간 동안 시간이 경과 한 이후에 시작할 경우에는 [+상대시간, 수행기간]로 표기한다. 수행기간이 0일 경우 특별한 실행 기간을 지정하지 않은 것을 의미한다. 이러한 기능은 공격자에 의해 보다 정밀하게 조정된 공격을 가능하게 하고, 또한 특정 시간에 동작을 시작하는 분산 서비스 거부 공격이나 바이러스, 웜 공격을 표현할 수 있도록 해 준다.



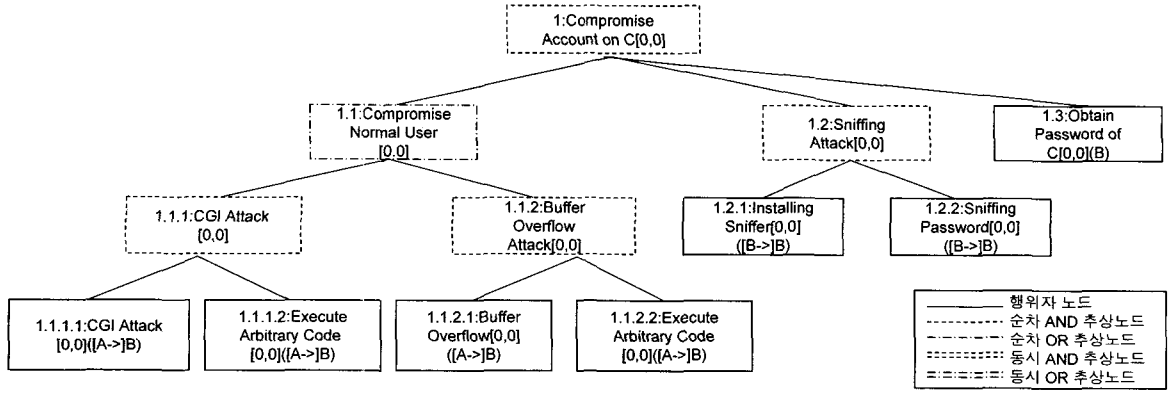
(그림 4) 노드 표기법

(그림 4)는 개선된 트리 기반 모델링 기법에서의 세분화된 노드들의 표기법을 보여준다.

(그림 5)와 (그림 6)은 (그림 2), (그림 3)에서 보인 그래프 기반 공격 모델링 기법과 트리 기반 공격 모델링 기법에서 사용한 공격 시나리오를 각각 개선된 모델링 표기법으로 기술한 것이다. 두 그림에서 그래프 기반 공격 모델링 기법과 트리 기반 공격 모델링 기법의 한계점으로 지적되었던 실행 순서의 모호성이나 복잡한 공격 시나리오의 구성 문제가 해결되었음을 확인할 수 있다. 단계 번호를 사용함으로써 노드의 실행 순서가 명확하게 드러나게 되었고, 기능에 따른 노드의 구분으로 트리 기반 모델링 기법이 표현할 수 없었던 복잡한 시나리오가 개선된 모델링 기법에 의해 기술될 수 있음을 알 수 있다.



(그림 5) (그림 2)의 공격 시나리오 모델링



(그림 6) (그림 3)의 공격 시나리오 모델링

3.2 노드 기술 언어(Node Description Language)

공격 모델링에서 노드는 현실 세계에서 공격자가 실행하는 버퍼 오버플로우와 같은 공격 프로그램 이외에도 파일의 삭제, 복사, 변경 같은 명령어 실행이나 텔넷 프로그램, FTP와 같은 일반 프로그램의 실행 등을 모두 포함하는 공격자의 행위 전부를 의미한다. 이러한 공격자의 행위를 적절한 단위로 구분하여 노드로 표현해야 한다. 이때 각각의 노드는 행위자 노드와 추상 노드의 구분 없이 공통적으로 (그림 7)과 같은 규칙을 사용하여 표기한다.

노드 이름(node name)은 노드가 수행하는 공격을 직관적으로 표현할 수 있는 이름으로 표기한다. 만약 노드가 특정 공격 기술을 표현하는 것이라면, 기존의 공격 이름을 결정하는 규칙에 대한 연구 자료를 참조하여 명명할 수 있다 [7, 8]. 노드의 속성(property)에는 노드가 가지는 속성 정보를 표기한다. 행위자 노드의 예를 들면 공격의 CVE, CERT 아이디와 같은 취약성 번호, 공격의 종류, 관련 프로그램, 방어 방법 등과 같은 정보를 포함할 수 있다. 선조건(precondition)에는 공격이 수행되기 위한 조건을 점검하는 부분이다. 해당 프로그램의 설치 여부, 사용자 권한, 파일의 존재 유무, 네트워크 상태 등이 여기에 속한다. 실행(action)은 실제 공격을 수행하는 부분이고 이 부분은 공격의 종류에 따라 수행하는 내용이 많이 달라진다. 후조건(postcondition)은 실행 부분에서 수행된 결과를 점검하고 전체 노드의 수행 결과를 알려주는 역할을 한다.

```
Node Name : node_name [          // 노드
    Property[item_list]          // 노드의 속성
    Precondition[item_list]      // 노드의 수행 조건
    Action[item_list]            // 노드의 수행
    Postcondition[item_list]     // 노드의 수행 결과
]

node_name                        // 노드의 이름
item_list                        // 노드의 속성 리스트
```

(그림 7) 노드의 기술 규칙

```
Node Name : Wu-Ftpd Remote Format String Stack Overwrite Vulnerability
[
    Property [
        Step = Undecided
        CVE ID = CVE-2000-0573
        Bugtraq ID = 1387
        Class = Input Validation Error
        Related Software = Wu-ftpd 2.4.2 - 2.6.0
        Reference = http://online.securityfocus.com/bin/1387/
        Start Time = Undecided
        Operation Time = Undecided
    ]

    Precondition [
        User Privilege = Normal or Root
        Network = Online
    ]

    Action[
        Step 1. Connect target machine
             2. Generate Invalid Input and Arbitrary Code
             3. Send Invalid Input with Arbitrary Code
    ]

    Postcondition[
        Step 1. Check Attack Result
             2. Return Result
    ]
]
```

(그림 8) Wu-ftpd 포멧 문자열 공격 노드의 기술

Wu-FTP 프로그램은 워싱턴 대학에서 개발된 인기 있는 유닉스 파일 교환 프로그램이다. 이 프로그램은 특정 버전에서 잘못된 명령어 문자열이 입력될 경우 악성 코드를 실행하는 포멧 문자열 공격(format string attack)에 취약성을 가진다. 이 공격을 (그림 8)과 같이 앞서 설명한 노드 기술 언어를 이용해 표현할 수 있다. 그림에서 속성으로 취약성 아이디와 Bugtraq ID는 각각 CVE-2000-0573, 1387 이고, 공격의 종류는 잘못된 입력 에러(input validation error)이다. 관련 소프트웨어는 Wu-ftp 2.4.2 버전에서 2.6.0 이다. 다만 단계번호, 시작시간, 수행기간은 공격 시나리오

구성 시에 결정되므로 초기 상태에서는 미정이다. 선조건으로 네트워크에 연결된 컴퓨터의 일반 사용자 또는 루트 권한이 필요하고, 공격의 수행은 목적 컴퓨터에 네트워크 연결, 악성 코드 및 잘못된 입력 문자열 생성, 생성된 악성 코드와 입력 문자열을 보내는 순서로 진행된다. 공격 종료 후 후조건에서 공격 결과를 점검하고 상위 노드에 공격 결과를 보낸다.

```
Node Name : Undecided
[
    Property [
        Step = Undecided
        Substeps = undecided
        Order = Sequential
        Operation = AND
        CVE ID = None
        Bugtraq ID = None
        Class = None
        Related Software = None
        Reference = None
        Start Time = Undecided
        Running Time = Undecided
    ]

    Precondition [
        None
    ]

    Action[
        Run substeps sequentially
    ]

    Postcondition[
        Step 1. Check Attack Result
             2. Return Result
    ]
]
```

(그림 9) 순차 AND 추상 노드의 기술 예

추상 노드의 경우에는 속성, 선조건, 실행, 후조건의 노드 기본 구성은 같지만 내부적으로 다른 부분을 필요로 한다. (그림 9)의 순차 AND 추상 노드의 사용 예를 보이고 있다. 행위자 노드와는 약간의 차이가 있음을 알 수 있다. 먼저, 속성 부분에서 노드 이름은 시나리오 구성 시에 하위 노드들의 특성을 고려해서 작명해야 하므로 초기에는 미정인 상태이다. 또한 CVE와 Bugtraq ID 그리고 참고 자료는 기술되지 않는데, 추상 노드의 특성상 특별한 공격을 지정하지 않기 때문이다. 새로이 추가된 속성으로 하위 노드(substep), 순서(order), 연산(operation)이 있다. 하위 노드는 시나리오 구성 시에 추상 노드가 자신의 하위 노드들에 대한 단계 번호를 기술하게 된다. 순서는 하위 노드의 단계 번호에 따른 순차적 실행을 의미하고, 연산은 하위 노드의 수행 결과를 종합해서 추상 노드 자신의 전체 수행 결과의 성공 실패 여부를 결정하는데 사용하게 된다. 선조건은 특별히 지정되지 않으며, 실행에서는 하위 노드를 순차적으로 실행한다. 후조건에서는 실행 부분의 결과를 종합하고, 속성에 지정된 연산에 따라 추상 노드 자신의 수행 결과를 결정하는 부분이다. 다른 세가지 추상 노드인 순차 OR, 동시 AND, 동시 OR 추상 노드 등도 위에서 설명된 순차 AND 추상 노드와 거의 동일한 구조를 가지며, 실행 순서와 실행 결과에 따른 연산 부분만이 각 추상 노드의 특성을 반영한다.

3.3 노드 재사용 및 확장

실 세계에서는 하루가 다르게 새로운 공격 방법이 생겨나고 있다. 공격 방법은 기존에 이용되지 않던 전혀 새로운 방식의 공격도 있지만, 많은 공격들이 기존의 공격 방법을 조금 수정하거나 또는 여러 기존 공격들을 조합해서 새롭게 만들어 진다. 또한 아무런 공격 의도를 포함하지 않는 정상적인 명령어도 적절한 조합을 통해 새로운 공격으로 생성될 수 있다.

```
Node Name : Nmap
[
    Property [
        Step = Undecided
        Substeps = undecided
        Order = Sequential
        Operation = AND
        CVE ID = None
        Bugtraq ID = None
        Class = Scanning
        Related Software = None
        Reference = None
        Start Time = Undecided
        Running Time = Undecided
    ]

    Precondition [
        User Privilege = Root
        Network = Online
        Nmap = Installed
    ]

    Action[
        Step 1. Host Scanning Node
             2. Port Scanning Node
             3. OS Fingerprint Scanning Node
    ]

    Postcondition[
        Step 1. Check Attack Result
             2. Return Result
    ]
]
```

(그림 10) 기존 노드들을 재사용한 Nmap 공격 노드의 기술

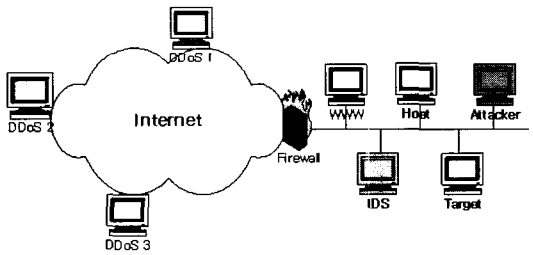
Nmap과 같은 스캐닝 공격(scanning attack) 프로그램을 고려해 보자. 스캐닝 공격은 호스트의 온라인 접속 여부를 확인해 주는 호스트 스캐닝(host scanning), 호스트가 제공하는 서비스 포트를 살펴보는 포트 스캐닝(port scanning), 호스트가 운영하는 소프트웨어나 운영체제 관련 정보를 알

려 주는 핑거프린팅 스캐닝(fingerprinting scanning)등의 세 가지 공격으로 구분된다. 기존의 스캐닝 공격 도구들은 한두 가지의 스캐닝 기능만을 제공했으나, Nmap은 기본적으로 세 가지 스캐닝 기능을 모두 제공하는 공격 도구이다. 만약 기존에 이미 호스트, 포트, 핑거프린팅 스캐닝과 같은 세 가지 스캐닝 노드가 구현되어 있다면, 이 노드들을 사용하여 Nmap과 같은 새로운 공격 노드를 표현하는 것이 가능해 진다. 이는 곧 기존 노드의 재사용성과 확장성을 지원하여 사용자에게 편의를 제공할 수 있음을 의미한다.

새로이 개선된 공격 모델링 기법에서는 추상 노드를 활용함으로써 노드의 재사용 및 확장성을 쉽게 지원할 수 있다. (그림 10)은 추상 노드와 미리 만들어진 공격 노드들을 이용해 Nmap 공격 노드를 구성한 모습이다. 속성 부분에 필요한 부분을 설정하고, 선조건 부분에서 실행 권한, 네트워크 연결 여부, 프로그램 설치 여부를 확인하고, 실행 부분에서 순차적으로 호스트, 포트, 핑거 프린팅 스캐닝을 실행하는 것을 확인할 수 있다.

4. 개선된 모델링 기법의 검증

본 절에서는 개선된 모델링 기법의 타당성을 검토하기 위해 보다 복잡한 공격 시나리오를 개선된 공격 모델링 기법을 바탕으로 작성하여 본다. 이 시나리오는 기존 모델링 기법에서 표현할 수 없었던 동시에 여러 공격이 수행되는 경우와 각 공격의 시작 및 수행기간이 공격자의 의도에 의해 조정되는 것들을 포함한다.



(그림 11) 네트워크 구성

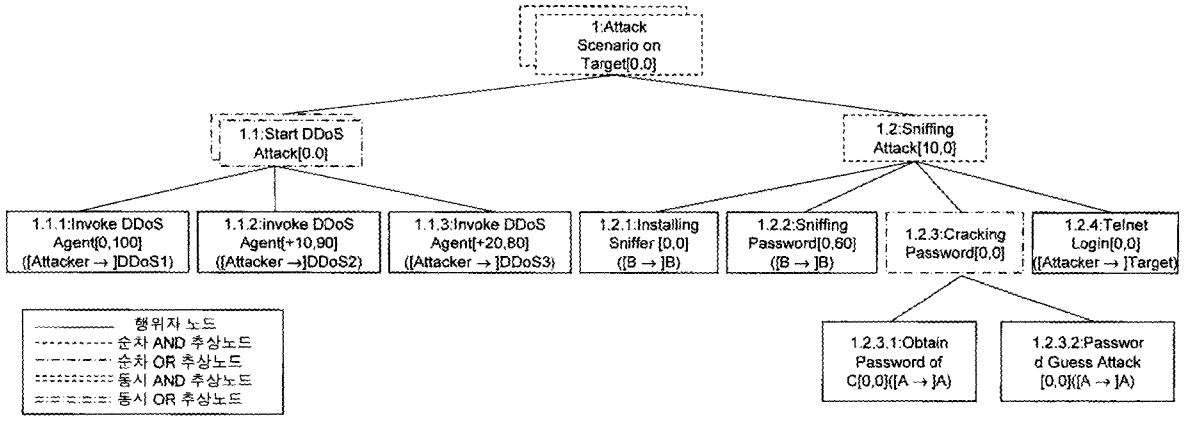
(그림 11)은 어떤 기업의 네트워크 구성도로서 공격 시나리오를 위한 기반 네트워크 형태를 보여준다. 방화벽으로 내부 네트워크를 보호하고, 방화벽 뒤에 다시 침입 탐지 시스템(IDS)을 설치하여, 공격에 대비하고 있다. 이 회사는 웹 서버(WWW)를 운영하고 있으며, 공격자(attacker)와 타겟 시스템(target), 호스트 그리고 침입 탐지 시스템은 같은 랜 영역 안에 있다.

(그림 12)는 (그림 11)의 네트워크 구성에 대한 세부 네트워크 설정을 보여준다. 그림에서 방화벽의 경우 외부로 나가는 패킷은 모두 허용하고 내부로 들어오는 패킷에 대해 80번 포트를 제외한 1-1024 포트에 대해 필터링 한다. 침입 탐지 시스템은 서비스 거부 공격에 대한 탐지 룰과 스니핑 공격에 대한 탐지 룰을 가지고 있다. 그리고 최종 목표가 되는 타겟 시스템은 텔넷 서비스를 하고 있다는 점에 유의하기 바란다.

공격 시나리오는 다음과 같다. 어떤 기업에 근무하는 공격자는 기업의 인사 정책에 불만을 품고 기업 내부 정보를 탈취하기 위해 타겟 시스템의 계정을 얻는 것을 목표로 하고 있다. 이를 위해 스니핑 공격을 이용하여 타겟 시스템으로 로그인하는 계정을 훔치려고 한다. 공격자는 침입 탐지 시스템이 스니핑 공격을 탐지하는 것을 방해하기 위해 미리 외부 네트워크에 설치해 놓은 서비스 거부 공격(DoS) 툴을 실행시켜 회사의 웹 서버를 공격한다. 이 공격에서 중요한 점은 분산 서비스 거부 공격(DDoS)과 스니핑 공격은 동시에 수행되어야 하지만, 스니핑 공격은 분산 서비스 거부 공격 이후 어느 정도 시간이 경과되어 침입 탐지 시스템에 충분한 부하가 걸리는 시점에서 시작하고, 분산 서비스 거부 공격이 끝나는 시점 전에 스니핑 공격이 마무리되어야 공격이 성공적으로 끝날 수 있다는 점이다.

```
WWW [
        OS = Microsoft Window 2000
        HTTP Server = Microsoft IIS 5.0
]
Target [
        OS = FreeBSD 4.3
        Telnet Server = BSD Telnet Server
        Account  : [abc : xyz]
]
Host [
        OS = Redhat Linux 6.0
        Telnet Client = Linux Telnet Client
]
Attacker[
        OS = Redhat Linux 7.0
        Telnet Client = Linux Telnet Client
        Software = Sniffer
]
IDS[
        Rule 1[Detect DoS Attack]
        Rule 2[Detect Sniffing Attack]
]
Firewall [
        Rule 1[Egress all Accept]
        Rule 2[Ingress port 80 Accept]
        Rule 3[Ingress port 1-1024 Deny]
        Rule 4[Ingress all Accept]
]
DDoS 1[
        OS = Redhat Linux 6.0
        Software = DDoS Agent Installed
]
DDoS 2[
        OS = Microsoft Window 2000
        Software = DDoS Agent Installed
]
DDoS 3[
        OS = Microsoft Window 2000
        Software = DDoS Agent Installed
]
```

(그림 12) 네트워크 설정



(그림 13) 개선된 모델링 기법을 적용한 공격 시나리오 모델링

(그림 13)은 앞서 기술한 공격 시나리오를 개선된 트리 기반 공격 모델링 기법으로 구성한 것이다. 그림에서 설명하는 공격 시나리오 모델은 시뮬레이션의 시작과 동시에 단계 1의 동시 AND 추상 노드가 실행되고, 이 노드는 자신의 하위 노드 1.1, 1.2을 동시에 수행한다. 노드 1.1은 자신의 하위 노드로 분산 서비스 공격을 동시에 실행하는데, 1.1.1은 시작시간이 0이므로 바로 수행하여 100초 동안, 1.1.2는 10초 후 수행하여 90동안, 1.1.3은 20초 후 실행하여 80초 동안 서비스 거부 공격을 수행한다. 따라서 1.1 노드는 총 100초 동안 서비스 거부 공격을 수행한다. 1.1 노드가 실행된 10초 후 1.2 노드가 실행되는데, 이 노드는 순차 AND 추상 노드이므로 자신의 하위 노드를 단계 번호 순서대로 1.2.1 스니핑 프로그램 설치, 1.2.2 60초간 암호 스니핑 공격을 수행한 다음, 1.2.3 순차 OR 추상 노드를 수행한다. 이 노드는 1.2.2 스니핑 공격에서 얻어진 정보를 자신의 하위 노드 1.2.3.1이나 1.2.3.2에서 분석하여 암호 획득 또는 추측 작업을 수행한다. 이 모든 작업이 성공적으로 이루어진 경우 마지막 1.2.4 노드가 수행되면서 타겟 시스템에 로그인을 시도한다.

5. 결 론

본 논문에서는 시스템 보안 테스트, 네트워크 취약점 분석, 침입 탐지 시스템의 탐지 룰 표현을 위해 개발된 기존 인터넷 공격 모델링 기법을 보완하여 인터넷 보안 시뮬레이션을 위한 공격 모델링 기법으로 활용할 수 있게 개선하였다. 개선된 트리 기반 공격 모델링 기법은 복잡한 시나리오의 표현 불가, 실행 순서의 모호함, 시스템 상태 정보의 미 반영 등과 같은 기존 트리 기반 공격 모델링 기법의 문제점들을 해결하였다. 또한 기존 공격 모델링 기법들을 인터넷 보안 시뮬레이션용으로 사용하고자 할 경우 부적합했던 동시간 공격 표현, 정밀한 시작 및 수행기간 지정, 구현 편의성에 있어서도 대부분 해결될 수 있음을 확인했다.

노드의 재사용성을 높이기 위해서는 노드의 기본 단위를 정하는 문제가 중요하다. 본 논문에서는 노드의 기본 단위를 기존의 트리 기반 공격 모델링 기법에서 사용하던 것과 같이 공격 기술에서 공격자의 모든 행위(예, 파일을 복사하거나 프로그램을 실행하는 행위 등)로 정의하고 이를 바탕으로 모델링을 수행하였다. 그러나 "공격자의 행위"라는 의미는 모호해 질 수 있으며, 또한 이 방법이 재사용성을 위한 최선의 방법이라고 말 할 수는 없다. 따라서 재사용성을 높이기 위한 노드의 기본 단위에 대한 연구가 추가적으로 필요하다.

본 연구팀은 현재 실제 인터넷 공격을 시뮬레이션하기 위한 시뮬레이터를 개발 중이다. 이 시뮬레이터는 네트워크 시뮬레이션 도구인 SSFNet[15-16]을 기반으로 하고 있으며, 다양한 공격 시나리오를 기술하기 위해 본 논문에서 제시된 모델링 기법을 활용할 계획이다.

참 고 문 헌

[1] Donald Welch and Greg Conti, "A Framework for an Information Warfare Simulation," Proceedings of the 2001 IEEE, Workshop on Information Assurance and Security, United States Military Academy, West Point, NY, June, 2001.

[2] Shabana Razak, Mian Zhou and Sheau-Dong Lang, "Network Intrusion Simulation Using OPNET," Proceedings of OPNETWORK2002 Conference, Washington, USA, Sept., 2002.

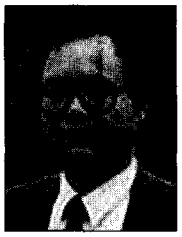
[3] T. Aslam, I. Krsul and E. Spafford, "Use of a Taxonomy of Security Faults," Proceedings of the 19th NIST-NCSC National Information Systems Security Conference, pp.551-560, 1996.

[4] S. Kumar, "Classification and Detection of Computer Intrusions," PhD Dissertation, Department of Computer Science, Purdue University, West Lafayette, Indiana, 1995.

[5] U. Lindqvist and E. Jonsson, "How to Systematically Classify Computer Security Intrusions," Proceedings of the IEEE Symposium on Security and Privacy, pp.154-163, 1997.
[6] J. Howard, "An Analysis of Security Incidents on the Internet, 1989~1995," PhD Dissertation, Department of Engineering and Public Policy, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1997.
[7] The MITRE Corporation, "Common Vulnerabilities and Exposures.", http://cve.mitre.org.
[8] The National Institute of Standards and Technology, "ICAT Metabase," http://icat.nist.gov.
[9] J. Mcdermott, "Attack Net Penetration Testing," In the New Security Paradigms Workshop (Ballycotton, County Cork, Ireland, Sept. 2000), ACM SIGSAC, ACM Press, pp.15-22, 2000.
[10] Paul Ammann, Duminda Wijesekera, and Saket Kaushik, "Scalable, graph-based network vulnerability analysis," Proceedings of the 9th ACM Conference on Computer and Communications Security, Washington, DC, USA, 2002.
[11] Jan Steffan, Markus Schumacher, "Collaborative Attack Modeling," Proceedings of the 2002 ACM Symposium on Applied Computing, Madrid, Spain, 2002.
[12] B. Schneier, "Attack Tree" Secrets and Lies. pp. 318-333, John Wiley and Sons, New York, 2000.
[13] T. Tidwell, "Modeling Internet Attack" Proceedings of the 2001 IEEE, Workshop on Information Assurance and Security, United States Military Academy, West Point, NY, June, 2001.
[14] Kristopher Daley, Ryan Larson, and Jerald Dawkins, "A Structural Framework for Modeling Multi-Stage Network Attacks," Proceedings of the 2002 IEEE, International Conference on Parallel Processing Workshops (ICPPW'02), Vancouver, B.c., Canada, August, 2002.
[15] James H. Cowie, "Scalable Simulation Framework API Reference Manual," Version 1.0 Document Draft - Revision, March, 1999, http://www.ssfnet.org
[16] SSF Research Network, "SSF Simulator Implementation," http://www.ssfnet.org/ssfImplementations.html.

서 정 국

e-mail : jkseo@cesys.ajou.ac.kr
2002년 아주대학교 정보 및 컴퓨터공학부 (학사)
2004년 아주대학교 정보통신전문대학원 정보통신공학과(공학석사)
관심분야 : 네트워크보안, 분산시스템 등



최 경 희

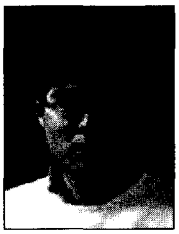
e-mail : khchoi@ajou.ac.kr
1976년 서울대학교 수학교육과(학사)
1979년 프랑스 그랑데꼴 Enseeiht 정보공학과(공학석사)
1982년 프랑스 Paul Sabatier 정보공학과 (공학박사)
1982년~현재 아주대학교 정보통신전문대학원 교수
관심분야 : 운영체제, 분산시스템, 실시간 및 멀티미디어시스템 등

정 기 현

e-mail : khchung@ajou.ac.kr
1984년 서강대학교 전자공학과(학사)
1988년 미국 Illinois주립대 EECS(공학석사)
1990년 미국 Purdue대학 전기전자공학부 (공학박사)
1991년~1992년 현대전자 반도체 연구소
1993년~현재 아주대학교 전자공학부 교수
관심분야 : 컴퓨터구조, VLSI 설계, 멀티미디어 및 실시간 시스템 등

박 승 규

e-mail : sparky@ajou.ac.kr
1974년 서울대학교 응용수학과(학사)
1976년 한국과학기술원 전산학과(공학석사)
1982년 프랑스 Institute National Polytechnique de Grenoble 전산학과(공학박사)
1976년~1992년 KIST, KIET, IBM 왓슨 연구소, ETRI 연구위원
1992년~현재 아주대학교 정보통신전문대학원 교수
관심분야 : 컴퓨터 구조, 멀티미디어, 실시간 시스템, 이동 컴퓨팅 등



심 재 홍

e-mail : jhshim@chosun.ac.kr
1987년 서울대학교 전산과학과(학사)
1989년 아주대학교 컴퓨터 공학과(공학석사)
2001년 아주대학교 컴퓨터공학과(공학박사)
1989년~1994년 서울시스템㈜ 공학연구소
2001년~2001년 아주대학교 정보통신전문대학원 BK 전임연구원
2001년~현재 조선대학교 인터넷 소프트웨어 공학부 전임강사
관심분야 : 운영체제, 분산 시스템, 실시간 및 멀티미디어 시스템 등