취약성 매트릭스를 이용한 사이버 공격 및 방어 모델링[〒*]

이장세[**], 지승도[***], 최규석[****]

Cyber Attack and Defense Modeling Using Vulnerability Metrics

Jang-Se Lee, Sung-Do Chi and Gyoo-Seok Choi

Abstract

The major objective of this paper is to perform modeling of cyber attack and defense using vulnerability metrics. To do this, we have attempted command level modeling for realizing an approach of functional level proposed by Nong Ye, and we have defined vulnerability metrics that are able to apply to DEVS(Discrete Event System Specification) and performed modeling of cyber attack and defense using this. Our approach is to show the difference from others in that (i) it is able to analyze behaviors of systems being emerged by interaction between functional elements of network components, (ii) it is able to analyze vulnerability in quantitative manner, and (iii) it is able to establish defense suitably by using the analyzed vulnerability. We examine an example of vulnerability analysis on the cyber attack and defense through case study.

Key Words: DEVS, Cyber Attack and Defense Modeling, Vulnerability Metrics

〒 본 논문은 과학기술부 한국과학재단 지정 경기도 지역협력 연구센터(RRC)인 한국항공대학교 인터넷 정보검색 연구센터(IRC)의 지원 및 청운대학교 교내 학술 연구비 지원에 의한 것임.
* 본 논문은 한국시뮬레이션학회 2003년 춘계 학술대회에서 발표한 내용을 수정, 보완한 것임.
** 한국해양대학교 IT공학부
*** 한국항공대학교 컴퓨터공학과
**** 청운대학교 컴퓨터학과

1. 서론

최근 들어 정보통신 시스템 자체의 버그, 부적절한 구성 설정, 개방형 인터넷 기반구조 등에 따른 취약성을 이용한 해킹의 피해가 급증하고 있다[1]. 이를 극복하기 위하여 정보통신 기반구조의 침해에 관한 보호와 더불어 신뢰성, 가용성 및 무결성을 보장하는 정보보증의 개념이 대두되고 있다. 이와 같은 정보보증을 위한 노력 중의 하나가 설계, 운영, 시험에 기준이 될 매트릭스를 설정하고, 이에 따른 적절한 모델을 개발하여 다양한 위협영향 평가 및 보안 대책 평가 등을 할 수 있는 모델링 및 시뮬레이션 기술의 개발이 필수적인 것으로 인식되고 있다[2,3]. 한편 네트워크 보안 모델링에 있어서 Cohen, Amoroso, Nong Ye등의 연구는 나름대로의 연구결과를 제시하고 있으나, Cohen[4]의 원인-결과 모델은 공격과 방어에 있어서 구체적인 상태의 변화나 동작 등을 분석할 수 없는 단점이 있다. 또한 Amoroso[5]의 침입탐지 모델 연구는 시뮬레이션 분석 및 활용에 대해서는 아직 미지수이며, Nong Ye[6]는 사이버 공격과 방어 모델에 대한 추상화 단계를 정함으로써 어떤 부분에 중점을 두어야 하는지에 대한 좋은 방향을 제시하고 있으나 모델링 및 시뮬레이션에 대한 구체적인 예시가 없는 실정이다.

따라서 본 논문에서는 명령어 레벨의 모델링을 통하여 Nong Ye가 제시한 기능적 단계의 접근을 시도하고, 이산사건 형식론인 DEVS (Discrete Event System Specification)에 적용 가능한 취약성 매트릭스를 정의하여 이를 이용한 사이버 공격과 방어에 대한 모델링을 수행한다. 본 연구는 기존 연구와 달리 첫째, 컴퓨터와 네트워크 시스템을 구성하는 각 구성요소들의 기능적인 요소와 이들 서로간의 상호작용에 의한 시스템의 동작을 구체적으로 분석할 수 있다. 둘째, 그에 따른 취약성을 정량적으로 분석할 수 있다. 셋째, 분석된 취약성을 이용하여 적절한 방어를 수립할 수 있다. 사례연구를 통하여 사이버 공격과 방어에 대한 취약성 분석의 예를 살펴본다.

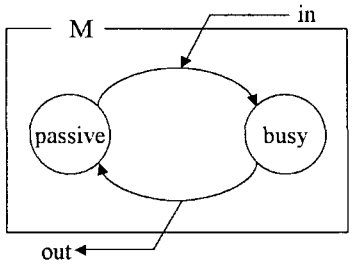
2. DEVS 형식론

이산 사건 모델링을 위한 대표적인 형식론인 DEVS는 연속적인 시간상에서 이산적으로 발생하는 사건들에 대하여 시스템의 행위를 측정하는 것으로 다음과 같은 형식론에 의해 표현한다[7].

$M = < X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta >$

여기에서 X는 입력사건의 집합을 의미하고, S는 상태의 집합을 의미한다. Y는 출력의 집합을 의미하고 $\delta_{int}(\delta_{ext})$는 내부 상태전이 함수(외부 상태전이 함수)를 나타낸다. λ는 외부사건을 생성하는 출력함수를 의미하고 ta는 시간 진행 함수를 나타낸다.

예를 들어 주어진 시스템이 <그림 1>과 같은 상태전이를 할 경우, 입력사건의 집합 X에는 in이 포함되며, 상태의 집합 S에는 pssive와 busy가 포함된다. 출력집합 Y에는 out이 포함된다. passive 상태에서 외부로부터 in이 입력될 경우 외부 상태 전이함수 δ_{ext}(passive, e, in)에 의하여 busy로 상태전이가 이루어지며, busy 상태에서 일정 시간이 경과하며 내부 상태전이 함수 δ_{int}(busy)에 의하여 passive로 상태전이가 이루어진다. 이때 출력함수 λ(busy)에 의하여 out이 생성되며, 시간 진행 함수 ta(busy)는 일정한 시간이 된다.



<그림 1> 시스템의 태전이도

3. 사이버 공격 및 방어 모델링

Nong Ye는 [7]에서 컴퓨터와 네트워크 시스템에 대한 사이버 공격의 동작을 이해하고 분석하기 위하여 기능적 단계의 모델링에 대한 필요성을 강조하였다. 따라서 본 논문에서는 사이버 공격과 방어에 대한 모델링을 위하여 기능적 단계의 접근을 시도하였다.

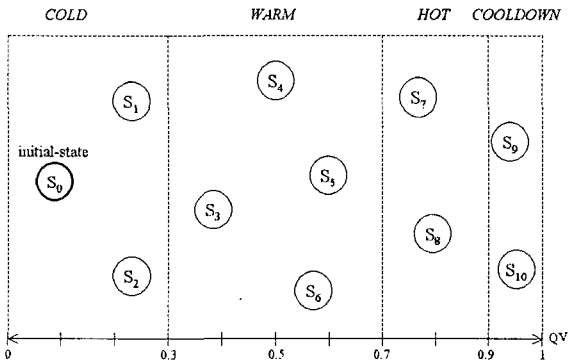
3.1. 시스템 상태

시스템의 상태는 시스템을 구성하는 다양한 자원의 상태에 대한 조합으로 정의하며, 시스템의 동역학은 DEVS 형식론에 의하여 표현된다.

$$S = \{s \mid s \in r_1 \times r_2 \times \cdots \times r_n\}$$

여기서 S는 시스템의 상태의 집합을 의미하여, r_i는 자원의 상태 집합을 의미한다.

<그림 2>는 DEVS 형식론에 의하여 표현될 수 있는 시스템의 상태 집합의 예를 나타낸다. <그림 2>에서 S_i는 공격 대상 호스트에 대한 각종 자원의 상태의 조합으로 구성된 시스템의 상태를 의미하며, 뒤에 설명되어질 취약성 값 QV(Quantitative Value)에 의하여 Wadlow[8]가 정의한 공격에 따른 네 가지의 상태로 구분할 수 있다.



<그림 2> 시스템의 상태 집합

3.2. 명령어

명령어는 시스템의 상태를 변환시키는 외부 입력 행위로서 정의한다. 즉 자원들의 현재 상태에 의존적으로 시스템의 해당 자원을 변경함으로써 현재의 상태를 다음의 상태로 변환시킨다.

$$C = \{c \mid s \xrightarrow{c} s', s, s' \in S\}$$

또한 연속적인 명령어에 의한 상태의 변환은 다음과 같이 축약할 수 있다.

$$s_0 \xrightarrow{c_0} s_1 \xrightarrow{c_1} \cdots \xrightarrow{c_n} s_n \Rightarrow s_0 \xrightarrow{x_{0,n}} s_n$$

여기서, $s_0, s_1, \ldots, s_n \in S$

$$x_{0,n} = \{c_0, c_1, \ldots, c_n \mid c_0, c_1, \ldots, c_n \in C\}$$

이와 같은 명령어 레벨의 모델링은 각 구성요소들의 기능적인 요소와 이들 서로 간의 상호작용에 의한 시스템의 동작을 구체적으로 분석하는 기능적 단계의 모델링과 시뮬레이션에 효과적으로 적용될 수 있다. 명령어 모델링은 명령어 실행을 위한 시스템의 현재 상태와 명령어 실행에 따른 시스템의 변경을 명세함으로써 이루어진다. 이를 위해서는 각종 서비스들에서 사용되는 명령어들의 그룹화 및 특성화가 수반되어야 한다[9].

<표 1> Telnet 명령어의 실행을 위한 선/후행 조건 테이블(일부 예)

Command	Pre-condition (current states)	Output	Post-condition (next states)
pwd	-	Return working directory name	-
rmdir	Check the file existance	Remove directory entries	Change directory attributes
cd	Check the file existance	Change working directory	Change directory attributes
chmod	Check the file existance	Change the permission mode	Change permission attribute

<표 1>은 Telnet 명령어에 대한 선/후행 조건 표현을 이용한 명령어 레벨의 모델의 예를 나타낸다. 여기서 'pre-condition'은 명령어가 실행되기 위한 조건에 대한 내용, 'output'는 명령어의 처리로 얻는 결과 내용, 그리고 'post-condition'은 명령어를 수행한 후에 변경되는 노드나 서비스의 속성에 대한 내용을 각각 나타낸다.

3.3. 취약성

취약성은 시스템의 정상적인 수행을 위배하기 위하여 공격되어질 수 있는 시스템의 약점으로서 자원들의 부적절한 상태로 정의한다. 또한 자원의 상태의 부적절한 정도와 공격에 의한 시스템의 영향 등을 고려함으로써 0 ~ 1 사이의 값으로 취약성을 정량화할 수 있다.

$$QV = f(s) \quad 단, 0 \le QV \le 1$$

여기서 QV는 정량적 취약성 값을 의미하며, f는 취약성 정량화를 위하여 취약성 매트릭스를 적용하는 함수를 의미한다.

취약성 매트릭스[10]는 네트워크상의 구성원들이 갖는 취약성 항목들에 대한 종합적인 취약성 값을 구하는 척도를 나타낸다. 이를 위하여 먼저 취약성 항목별 값의 범위를 0 ~ 1 사이의 값으로 정의하였는데, 이 값은 시뮬레이션 평가를 통하여 얻을 수 있다. 본 논문에서는 시뮬레이션 분석에 적용하기 위하여 <표 2>와 같이 취약성 항목을 크게 'Fixed Vulnerability'와 'Changeable Vulnerability'로 분류하였다. <표 2>에 나타난 바와 같이, 'Fixed Vulnerability'는 운영체제 종류, 버전, 서비스 종류 등과 같은 시스템 요소에 의존적인 취약성을 의미하며, 'Changeable Vulnerability'는 패스워드 구성 상태, 파일 접근 권한 등과 같은 시스템 구성 설정에 의존적인 취약성을 의미한다. 즉, 시뮬레이션 상에서의 다양한 시도와는 무관하게 취약성 값이 고정불변(즉, 네트워크 구성원의 상태변수 중 시간의 변화에 무관한 상태값)인 전자와는 달리 후자의 경우는 시뮬레이션을 통하여 다양한 취약성 값에 대한 분석을 수행할 수 있다. 'Impact level'은 해당 취약성을 이용한 공격 성공 시, 시스템에 미치는 영향을 나타내며 'Attack Scenario'는 해당 취약성을 이용하여 공격하기 위한 명령어의 집합으로 표현된다 (표 3 참조). 또한 이러한 공격에 대한 사전 대응방법으로 'Defense Strategy'를 정의할 수 있다.

<표 2> 구성원 취약성 테이블 (일부 예)

Category	Vul. item	Impact level (w_i)	Attack scenario #	Defense strategy	Remarks
Fixed vulnerability	Vul_{Phf}	0.75	scenario -2	web, telnet service protect	Phf CGI vulnerability
	Vul_{tmpfs}	1.0	scenario -5	O/S patch	SunOS 4.1.4 tmpfs vulnerability
	Vul_{glance}	1.0	scenario -4	O/S patch	HP-UXB 1.0.2 glance vulnerability
	...	...	...	...	...
Changeable vulnerability	$Vul_{Password}$	0.5	scenario -1 scenario -2	password managing periodically	Password vulnerability
	$Vul_{Userfile}$	0.75	scenario -3	file permission managing	User file vulnerability
	$Vul_{Filesystem}$	0.75	scenario -3	mountable file system managing	Files system vulnerability
	...	...	...	...	...

네트워크 상의 각 노드 구성원은 자신만의 구성 특성에 따라 <표 2>의 모든 취약성 항목들에 대한 부분집합을 갖는다. 따라서 노드 취약성은 해당 취약성 항목별 임팩트 레벨과 취약성 값의 곱에 대한 산술평균에 의하여 구할 수 있다. 즉, i번째 구성원의 노드 취약성인 NV_i를 다음과 같이 정의한다.

$$NV_i = \sum_{j=1}^{n}(w_j \times vul_j) / \sum_{j=1}^{n} w_j \quad (1)$$

여기서 n는 구성원에 대한 취약성 항목의 총 개수를 의미하며 w_j와 vul_j는 j번째 취약성 항목의 임펙트 레벨과 취약성 값을 의미한다.

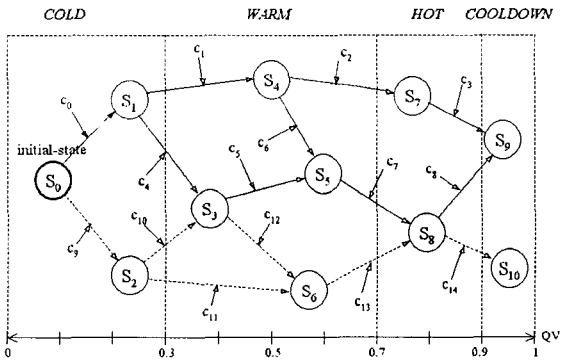
3.4. 사이버 공격 모델링

공격은 비인가된 상태로 전이하기 위하여 취약성을 이용하여 자원을 부정하게 변경하는 연속적인 행위로 모델링한다.

$$A = \{a \mid s_0 \xrightarrow{a_{0,n}} s_n,\ s_0,\ s_n \in S\}$$

이때, $QV(s_{n-1}) < QV(s_n)$, 단 $n \geq 1$

여기서 A는 공격의 집합을 의미하여, s_0는 초기상태를 의미한다. s_n은 설정한 QV를 갖는 비인가 된 상태(COOLDOWN)를 의미한다.



<그림 3> 시스템 상태 전이도
(사이버 공격의 경우)

<그림 3>은 DEVS 형식론에 의하여 표현될 수 있는 알려진 공격과 알려지지 않은 공격에 대한 시스템의 상태 전이도를 나타내며 화살표는 상태변환을 뜻한다. 그림 3에서 실선은 알려진 변환을 나타내며, 초기상태인 s_0로부터 비인가 된 상태인 s_9에 이르게 하는 일련의 명령어의 집합 $a_1 = \{c_0, c_1, c_2, c_3\}$, $a_2 = \{c_0, c_1, c_6, c_7, c_8\}$, $a_3 = \{c_0, c_4, c_5, c_7, c_8\}$은 알려진 공격을 나타낸다. 한편 점선은 알려지지 않은 변환을 나타내는데, 초기상태인 s_0로부터 비인가된 상태인 s_{10}에 이르게 하는 $a_4 = \{c_9, c_{10}, c_{12}, c_{13}, c_{14}\}$와 $a_5 = \{c_9, c_{11}, c_{13}, c_{14}\}$는 알려지지 않은 공격을 나타낸다.

<표 3>에서 'Attack scenario #'는 공격 시나리오의 번호를 나타내고, 'Commands'는 연속적인 명령어들의 순차적 리스트를 나타낸다. 'Exploited vulnerability'는 공격을 위하여 사용된 취약성 항목을 의미하여 'Effected vulnerability'는 공격 성공에 의하여 변화되는 취약성 항목을 나타낸다.

<표 3> 공격시나리오 (일부예)

Attack scenario #	Commands	Exploited vul.	Effected vul.
1	telnet target.host Brute force passwords	$Vul_{Password}$	-
2	http://Node-3/cgi-bin/phf.cgi Brute force passwords	Vul_{Phf} $Vul_{Password}$	-
3	showmount -e target.host mount target.host:/usr/tmp cd /tmp echo abcxyz:1234:10001:1::: >> passwd su abcxyz echo attacker >> .rhosts rlogin attacker	$Vul_{Filesystem}$ $Vul_{Userfile}$	$Vul_{Filesystem}$ $Vul_{Password}$ $Vul_{Userfile}$
...	..	...	...

3.5. 방어 모델링

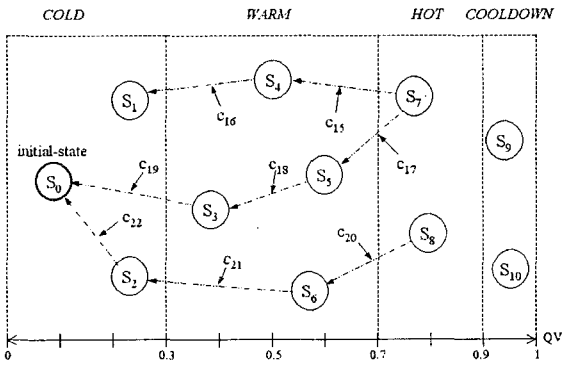
방어는 시스템을 안전한 상태로 유지하기 위하여 시스템을 구성하는 자원을 변경하는 연속적인 행위로 모델링한다.

$$D = \{d \mid s_n \xrightarrow{d_{n,0}} s_0,\ s_0,\ s_n \in S\}$$

이때, $QV(s_{n-1}) < QV(s_n)$, 단 $n \geq 1$

여기서 D는 방어의 집합을 의미하며, s_n는 방어 적용을 위한 임계치 값으로 설정한 QV를 갖는 상태(HOT)를 의미한다. s_0는 설정한 QV를 갖는 안전한 상태(COLD)를 의미한다.

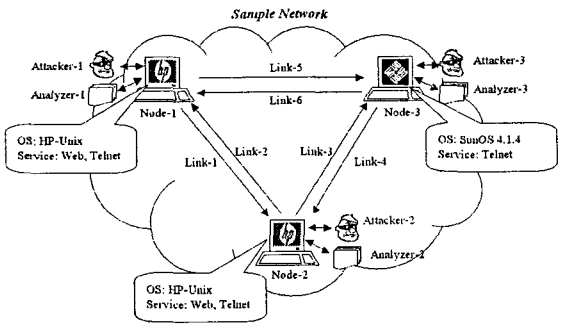
또 <그림 4>는 DEVS 형식론에 의하여 표현될 수 있는 방어의 경우에 대한 시스템의 상태 전이도를 나타낸다. 그림 4에서 이중점선은 방어를 나타내는 것으로서, 방어 적용을 위하여 설정한 임계치(QV = 0.7이상)에 도달한 상태로부터 안전한 상태(QV = 0.3이하)로의 전이를 위한 일련의 명령어의 집합이다. 이 경우 방어는 $d_1 = \{c_{15}, c_{16}\}$, $d_2 = \{c_{17}, c_{18}, c_{19}\}$, $d_3 = \{c_{21}, c_{22}, c_{23}\}$이 된다.



<그림 4> 시스템 상태 전이도(방어의 경우)

4. 사례연구

<그림 5>와 같은 세 개의 노드로 구성된 간단한 네트워크에 대한 모델링 및 시뮬레이션을 통하여 사이버 공격과 방어에 따른 취약성 분석을 수행하였다.



<그림 5> 샘플 네트워크

<그림 5>에 표현된 바와 같이, Node-1과 Node-2는 HP-Unix 운영체제를 기반으로 각 노드에 대하여 Web, Telnet Service를 제공하고, Node-3는 SunOS 4.1.4를 기반으로 Node-2에 대하여 Telnet Service만을 제공한다고 가정한다.

사이버 공격과 방어에 대한 시뮬레이션을 위하여 각 노드의 Attacker 모델은 표 3의 공격 시나리오를 포함한 총 10개의 시나리오를 가지고 상대 노드에 대한 공격을 수행한다. 각 노드별 취약성 초기 값은 각 노드에 설치된 운영체제, 제공되는 서비스 및 파일 설정 등을 고려하여 표 4와 같다고 가정하자. 이 경우 (case1의 'static'), Node-1의 노드 취약성은 표 2의 임팩트 레벨(w_i)과 표 4의 초기값(vul_i)을 식(1)에 적용함으로써 구할 수 있다. 따라서 Node-1의 노드 취약성인 NV_{node-1}은 (0.5 × 0.5 + 0.75 × 0.25 + 0.75 × 0.5 + 0.75 × 1.0 + 1.0 × 1.0 + 1.0 × 0.0) / (0.5 + 0.75 + 0.75 + 0.75 + 1.0 + 1.0) = 0.54가 되며 Node-2와 Node-3의 노드 취약성인 NV_{node-2}, NV_{node-3}는 각각 0.5, 0.32가 된다.

<표 4> 구성원 모델의 초기조건

Component name	Vulnerability items(vul_i)	
	Changeable items	Fixed item
Node-1	$Vul_{Password}$: 0.5 $Vul_{Filesystem}$: 0.25 $Vul_{Userfile}$: 0.5	Vul_{Pht} : 1.0 Vul_{glance} : 1.0 Vul_{tmps} : 0.0
Node-2	$Vul_{Password}$: 0.5 $Vul_{Filesystem}$: 0.25 $Vul_{Userfile}$: 0.25	Vul_{Pht} : 1.0 Vul_{glance} : 1.0 Vul_{tmps} : 0.0
Node-3	$Vul_{Password}$: 0.25 $Vul_{Filesystem}$: 0.0 $Vul_{Userfile}$: 0.5	Vul_{Pht} : 0.0 Vul_{glance} : 0.0 Vul_{tmps} : 1.0

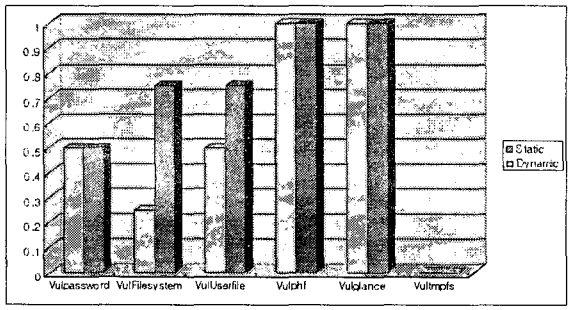
또한 <표 5>는 사이버 공격을 이용한 시뮬레이션을 통하여 얻어진 각 노드의 취약성 변화에 대한 결과로서 <표 4>에서와 같은 초기 설정에 따른 취약성에 의하여 각 노드에 대한 공격이 이루어지게 된다. 'remark'에 나타난

것과 같이 각 노드의 변화 가능한 취약성 항목에 대한 취약성이 증가하여 이때의 NV_{node-1}, NV_{node-2}, NV_{node-3}는 각각 0.66, 0.61, 0.45로 증가된다.

<표 5> 시뮬레이션 결과: 취약성 테이블(case1의 'dynamic')

Name	Vulnerability	Remark
Node-1	0.5x0.5 + 0.75x0.75 + 0.75x0.75 + 0.75x1.0 + 1.0x1.0 + 1.0x0.0 / (0.5+0.75+0.75+0.75+1.0+1.0) = 0.66	$Vul_{Filesystem}$, $Vul_{Userfile}$ are increased
Node-2	0.5x0.5 + 0.75x0.7 + 0.75x0.5 + 0.75x1.0 + 1.0x1.0 + 1.0x0.0 / (0.5+0.75+0.75+0.75+1.0+1.0) = 0.61	$Vul_{Password,,}$ $Vul_{Filesystem}$, $Vul_{Userfile}$ are increased
Node-3	0.5x0.5 + 0.75x0.5 + 0.75x0.7 + 0.75x0.0 + 1.0x0.0 + 1.0x1.0 / (0.5+0.75+0.75+0.75+1.0+1.0) = 0.45	$Vul_{Password,,}$ $Vul_{Filesystem}$, $Vul_{Userfile}$ are increased

<그림 6>은 Node-1에서의 개별 취약성의 변화를 나타낸다. 그림에서 'static'은 시뮬레이션 분석 전에 네트워크의 현재 상태에 따른 고정적인 취약성 값만을 의미한다. 반면 'dynamic'은 고정적인 취약성과 더불어 시뮬레이션을 수행함으로써 분석된 변경 가능한 취약성까지 고려한 취약성 값을 의미하는 것으로서 취약성 항목들의 값이 증가하는 것을 알 수 있다. 이와 같이 현재 노드의 상태에 따른 취약성뿐만 아니라 사이버 공격에 대한 시뮬레이션을 통하여 향후 발생 가능한 취약성을 사전에 분석할 수 있다. 또한 노드에 대한 취약성과 더불어 개별 취약성 항목들에 대한 분석이 가능함으로써 노드를 구성하는 자원들에 대한 구체적인 방어를 결정하는데 활용할 수 있다.



<그림 6> 개별 취약성의 변화 (Node-1의 경우)

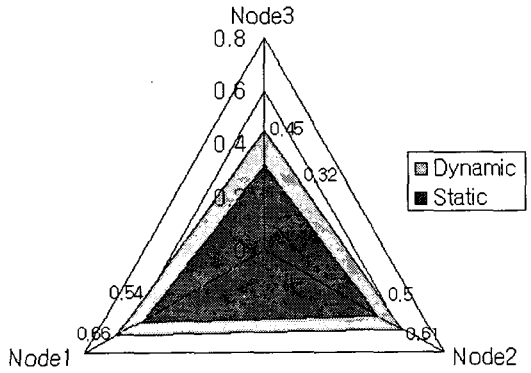
예를 들어, <표 2>의 'Defense Strategy'에 따라 Node-2는 운영체제를 패치하는 방어를 시행하고 Node-3는 파일의 권한 설정을 변경하는 방어를 시행할 수 있다. 이 경우(case2의 'static'), Node-2는 고정된 취약성 항목 중 Vul_{glance}의 값이 1.0에서 0.0으로 감소되며 Node-3는 변화 가능한 취약성 항목 중 $Vul_{Userfile}$의 값이 0.5에서 0.0으로 감소됨으로써 NV_{node-1}, NV_{node-2}, NV_{node-3}는 각각 0.54, 0.29, 0.24이 된다.

또한 <표 6>은 이와 같은 방어를 적용하여 시뮬레이션 한 결과로서, 이때의 NV_{node-1}, NV_{node-2}, NV_{node-3}는 각각 0.66, 0.31, 0.27로 감소함을 알 수 있다.

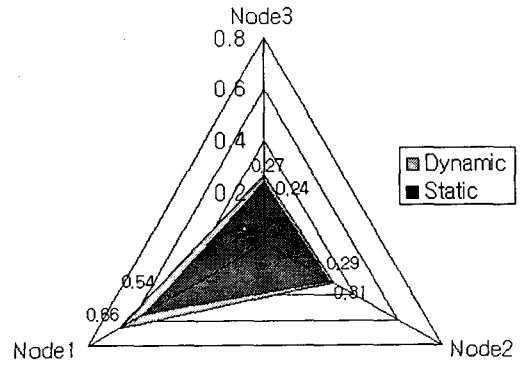
<표 6> 시뮬레이션 결과: 취약성 테이블 (case2의 'dynamic')

Name	Vulnerability	Remarks
Node-1	0.5x0.5 + 0.75x0.75 + 0.75x0.75 + 0.75x1.0 + 1.0x1.0 + 1.0x0.0 / (0.5+0.75+0.75+0.75+1.0+1.0) = 0.66	$Vul_{Filesystem}$ and $Vul_{Userfile}$ are increased
Node-2	0.5x0.6 + 0.75x0.28 + 0.75x0.3 + 0.75x1.0 + 1.0x0.0 + 1.0x0.0 / (0.5+0.75+0.75+0.75+1.0+1.0) = 0.31	Node vulnerability is not increased
Node-3	0.5x0.27 + 0.75x0.1 + 0.75x0.12 + 0.75x0.0 + 1.0x0.0 + 1.0x1.0 / (0.5+0.75+0.75+0.75+1.0+1.0) = 0.27	$Vul_{Filesystem}$ and $Vul_{Userfile}$ are increased

<그림 7>은 샘플 네트워크에 대한 취약성 변화 추이를 나타낸다. 그림에서 'static'은 시뮬레이션 분석 전에 네트워크의 현재 상태에 따른 고정적인 취약성 값만을 의미하는 반면, 'dynamic'은 고정적인 취약성과 더불어 시뮬레이션을 수행하여 변경 가능한 취약성까지 고려한 취약성 값을 의미한다. 앞서 그림 6에 나타난 바와 같이 노드를 구성하는 개별 취약성 항목들에 대한 동적 취약성이 증가함으로써 'static'에 비하여 'dynamic'의 취약성 값이 높게 나오는 것을 알 수 있다.



(a) Case 1: 방어 시행 전



(b) Case 2: 방어 시행 후

<그림 7> 샘플 네트워크의 취약성 변화 추이

<그림 7>(a)는 샘플 네트워크의 취약성 분석 결과를 나타내며, <그림 7>(b)는 <그림 7>(a)의 취약성 분석 결과를 토대로 방어 전략을 수립하여 시행한 후의 취약성 변화를 나타낸 것으로서 전체적으로 취약성이 감소하는 것을 알 수 있으며, 이를 통하여 적용된 방어가 적절함을 평가할 수 있다.

5. 결론 및 향후연구

본 논문은 취약성 매트릭스를 이용한 사이버 공격 및 방어에 대한 모델링을 주목적으로 하였다. Nong Ye가 제시한 기능적 단계의 접근을 위하여 명령어 레벨의 모델링을 시도하였으며, DEVS에 적용 가능한 취약성 매트릭스를 정의하여 사이버 공격과 방어에 대한 모델링을 수행하였다. 본 연구는 기존 연구와 달리 첫째 컴퓨터와 네트워크 시스템을 구성하는 각 구성요소들의 기능적인 요소와 이들 서로 간의 상호작용에 의한 시스템의 동작을 구체적으로 분석할 수 있으며, 둘째 그에 따른 취약성을 정량적으로 분석할 수 있다. 셋째 분석된 취약성을 이용하여 적절한 방어를 수립할 수 있는 장점을 갖는다. 향후 연구로서 네트워크 구성원에 대한 상세한 모델링이 요구되며 자동화된 코드 등을 이용한 다양한 사이버 공격에 대한 모델링 및 시뮬레이션 연구가 이루어져야 할 것이다.

참고문헌

[1] T.A. Longstaff, et al., "Are We Forgetting the Risks of Information Technology", *IEEE Computer*, pp 43-51, Dec. 2000.

[2] DoD, *Defensive Information Operations, Technical Report #6510.01B*, June, 1997.

[3] A. Jones, "The challenge of building survivable information intensive systems", *IEEE Computer*, August, pp. 39-43, 2000..

[4] F. Cohen, "Simulating Cyber Attacks,

Defenses, and Consequences", *Proc. of IEEE Symposium on Security and Privacy Special 20th Anniversary Program*, Berkeley, CA, 1999.

[5] E. Amoroso, *Intrusion Detection*, AT&T Laboratory: Intrusion Net Books, 1999.

[6] N. Ye, and J. Giordano, "CACS - A Process Control Approach to Cyber Attack Detection", *Communications of the ACM*, Vol.44(8), 2001.

[7] B.P. Zeigler, *Object-oriented Simulation with Hierarchical, Modular Models: Intelligent Agents and Endomorphic Systems*, Academic Press, 1990.

[8] T.A. Wadlow, *The Process of Network Security*, Addison-Wesley, 2000.

[9] S.D. Chi, et. al., "Network Security Model- ing and Cyber-attack Simulation Metho- dology", Lecture Notes on Computer Science series, *6th Australian Conf. On Information Security and Privacy*, Sydney, Jul. 2001.

[10] J.S. Lee, J.R. Jung, and S.D. Chi, "Vulner ability Measures for Network Vulner ability Analysis System", *Proc. of 2002 IRC International Conference on Internet Information Retrieval*, Korea, Nov. 2002.

주 작 성 자 : 이 장 세
논 문 투 고 일 : 2003. 10. 30
논 문 심 사 일 : 2004. 07. 16(1차), 2004. 07. 19(2차), 2004. 07. 22(3차)
심 사 판 정 일 : 2004. 07. 22

● 저자소개 ●

이장세

1997 한국항공대학교 전자계산학과 학사
1999 한국항공대학교 컴퓨터공학과 석사
2003 한국항공대학교 컴퓨터공학과 박사
2004~현재 한국해양대학교 IT 공학부 전임강사
관심분야 : 모델링 및 시뮬레이션, 네트워크 보안, 지능시스템 설계, 인공생명

지승도

1982 연세대학교 전기공학과 학사
1984 연세대학교 전기공학과 석사
1985~1986 두산 컴퓨터 (현 한국 디지털) 근무
1991 미국 아리조나대학교 전기전산공학과 박사
1991~1992 미국 SIMEX Systems and S/W 회사 S/W 담당자로 근무
1992~현재 한국항공대학교 컴퓨터공학과 교수
관심분야 : 이산사건 시스템 모델링 및 시뮬레이션, 컴퓨터 보안, 지능시스템 디자인 방법론, 시뮬레이션 기반 인공생명, 교통 모델링

최규석

1982 연세대학교 전기공학과 학사
1987 연세대학교 전기공학과 석사
1987~1990 (주) 데이콤 중앙연구소 연구원
1991~1997 SK 텔레콤 책임연구원
1997 연세대학교 전기공학과 박사
1997~현재 청운대학교 컴퓨터학과 조교수
관심분야 : 인공지능 및 경로탐색, 유전 알고리즘, 이동통신