디지털산업정보학회 논문지 제6권 제3호-2010년 9월

네트워크 취약성 분석을 위한 확장된 사이버 공격 트리에 관한 연구*

엄 정 호* · 박 선 호** · 정 태 명***

A Study on an Extended Cyber Attack Tree for an Analysis of Network Vulnerability

Eom, Jung Ho · Park, Seon Ho · Chung, Tai M.

〈Abstract〉

We extended a general attack tree to apply cyber attack model for network vulnerability analysis. We defined an extended cyber attack tree (E-CAT) which extends the general attack tree by associating each node of the tree with a transition of attack that could have contributed to the cyber attack. The E-CAT resolved the limitation that a general attack tree can not express complex and sophisticate attacks. Firstly, the Boolean expression can simply express attack scenario with symbols and codes. Secondary, An Attack Generation Probability is used to select attack method in an attack tree. A CONDITION-composition can express new and modified attack transition which a aeneral attack tree can not express. The E-CAT is possible to have attack's flexibility and improve attack success rate when it is applied to cyber attack model.

Key Words : Attack tree, Extended Attack Tree, Cyber attack model

Ⅰ. 서론

네트워크 및 시스템 기술의 도약적인 발전으로 모든 기관이나 기업들이 정보통신체계 중심의 업무 프로세스를 확대하고 있다. 금융업무, 예약업무, 문서결재 등 모든 업무가 정보통신체계를 활용하여 이루어지고 있다. 그러나 그에 따른 역효과도 증가하고 있다. 사이버 공격자나 해커들은 경제적 이익이나 이직, 보복 등의 목적으로 정보통신체계를 마비시키거나 파괴시키고 그 속에 저장되어 있는 데이터를 변조, 유출하여 기관이나 기업에 큰 피해를 입히고 있다. 사이버 공격자들은 그들의 공격을 성공시키기 위하여 네트워크 시스템이 갖고 있는 고유한 취약점이나 해킹 프로그램을 사용하여 취약점을 식별한다[1-3].

이러한 취약점을 이용한 사이버 공격 방법이 정교하게 됨에 따라 취약점을 사전에 식별할 수 있는 취약점 분석 · 평가는 반드시 선행되어야 한다. 가장 효과적인 취약점 분석 · 평가 방법은 사이버 공격 방법이나 절차를 그대로 모방한 사이버 공격 모델을 설계하여 네트워크 시스템에 시뮬레이션하는 것이다[4-5].

사이버 공격 모델을 설계할 때 주로 공격 트리와 공격 그래프가 사용된다. 공격 트리는 보안 위협이나 실제 공격 절차를 설계하기 위한 분석적 기술로 많이 사용된다.

* 성균관대학교 정보통신학부 BK21 연구교수(제1저자, 교신저자)
** 성균관대학교 컴퓨터공학과 박사과정
*** 성균관대학교 컴퓨터공학과 교수

Bruce Schneier는 그의 논문[6]에서 공격 트리는 각 노드들의 자식 노드('AND'와 'OR' 노드)들이 실행이 완료되면 서브 목적을 달성할 수 있고 모든 서브 노드들의 실행이 완료되어 최종적으로 루트 노드가 실행되면 최종 공격목적을 달성하게 된다고 밝혔다. 공격 그래프 역시 사이버 공격절차 수립이나 취약점 분석을 위해 사용된다. Phillips와 Wei Wang은 그들의 논문[7-8]에서 공격 그래프를 공격자가 목적을 성공적으로 달성할 때까지 모든 침입경로를 묘사하였으며 공격 그래프의 노드는 공격상태를 나타내고 앳지는 공격상태 전이를 발생시키는 악의적인 활동들을 표현한 것이라고 밝혔다.

본 논문에서는 일반적인 공격 트리를 확장하여 사이버 공격 트리(E-CAT: Extended Cyber attack Tree)를 제안한다. 노드간의 조합(AND와 OR)에 공격을 강화할 수 있는 'CONDITION-composition(이하 CON)' 조합을 추가하였다. CON은 공격상황이나 공격대상 시스템 환경에 따라 'AND, OR-composition' 조합과 결합하여 새롭게 변형된 공격방법을 묘사할 수 있게 한다. 그리고 Boolean 표현식을 사용하여 서술식 공격 시나리오를 기호와 부호를 사용하여 간결하게 표현할 수 있게 하였다. 마지막으로 공격 트리 요소에 공격 생성확률(AGP: Attack Generation Probability, 이하 AGP) 요소를 추가하여 효과적으로 공격 경로를 선택하게 한다.

본 논문의 구성은 2장에서 관련 연구를 소개하고 3장에서 확장된 사이버 공격 트리를 제안하며, 4장에서 제안한 사이버 공격 모델을 평가한다. 마지막으로 5장에서 결론을 맺는다.

Ⅱ. 관련연구

공격 트리(Attack Tree)는 네트워크 시스템에 대한 다양한 공격에 대한 보안책을 수립할 수 있도록 전형적이고 방법론적인 해결책을 제시한다. 공격 트리 구조는 루트노드(Root Node)의 최종 공격목적과 중간노드(Sub Node)들의 목적을 달성하기 위한 다양한 공격 방법들을 묘사한다. 노드들은 일반적으로 선택 가능한 'OR' 노드와 목적을 달성하기 위해 반드시 수행해야 하는 'AND' 노드로 구성된다. 공격 트리가 설계되면 여러 중간노드들에게 가치를 할당하여 각각의 노드들의 가치를 계산하고 그 가치에 따라 보안대책을 수립하게 된다[9-10].

2.1 General Attack Tree

일반적인 공격 트리는 정점(ν), 앳지(ε), 조합(θ)의 요소로 이루어진다. 정점(ν)은 공격이나 공격목적을 나타내는 노드들의 집합이고 앳지(ε)는 자식노드에서 공격을 수행하여 부모노드로 이동하는 공격전이 상태들의 집합을 의미한다. 조합(θ)은 'OR, AND'로 구성되어 있는데 'OR' 조합으로 인접한 자식노드들은 그 중에 하나만 수행해도 공격목적을 달성할 수 있으며, 'AND' 조합으로 인접한 자식노드들은 모든 노드들이 수행되어야만 공격목적을 달성할 수 있다. 공격 트리의 표현식은 *Attack Tree = (ν, ε, θ)*와 같다.

① V는 서브공격이나 서브목적을 나타내는 노드들의 집합. $\nu \in V$는 공격자의 최종목적을 나타내는 트리의 최상의 루트노드를 나타낸다. V는 leaf_node와 internal_node로 구분된다.

(i) leaf_nodes ∪ internal_nodes = V

(ii) leaf_nodes ∩ internal_nodes = ∅, and

(iii) ν ∈ internal_nodes

② $\varepsilon \subseteq V \times V$는 공격 트리에서 앳지(edge)의 집합. *edge(u, v)* $\in \varepsilon$는 자식노드 v에서 부모노드 u까지 상태전이를 나타내는 *atomic attack*로 정의한다*(u, v ∈ V)*.

③ ε는 <ν, *composition*> 형식인 튜플들의 집합.

(i) ν ∈ internal_nodes and

(ii) composition ∈ [And-composition, OR-composition]

(iii) node ν ∈ *internal_nodes*가 만약 노드에 연결되어 있는 모든 앳지가 AND operation에 의해 연결되어 있으면 AND-composition으로 정의한다.

(iv) node $v \in internal_nodes$가 만약 노드에 연결되어 있는 모든 앳지가 OR operation에 의해 연결되어 있으면 OR-composition으로 정의한다.

2.2 Augmented Attack Tree

Nayot Poolsapassit는 'Investigating computer attacks using attack trees' 논문[11]에서 공격 트리를 이용하여 시스템 로그파일에서 공격과 연관된 정보만을 검출한 후 공격을 체계적으로 조사하는 디지털 포렌식 기술에 적용하였다. 논문에서 제시한 'Augmented attack tree'는 공격으로 간주할 수 있는 악의적인 행동 진행을 각각의 트리구조와 연관시킴으로써 공격을 탐지하게끔 한다. 즉, 시스템의 취약점 지식을 기반으로 로그 파일과 공격 트리를 비교함으로써 공격과 연관된 로그들만 필터링함으로써 다양한 공격경로를 식별한다.

Augmented attack tree의 형식은 다음과 같다.

정의 1. Atomic event는 *ordered pair <operation, target>*으로 구성된다.

정의 2. Atomic event가 시스템에 피해를 가하였을 경우에는 Atomic event를 Incident라 한다.

정의 3. Augmented attack tree는 $AAT = (V, E, \epsilon, Label, SIG_{u,v})$로 표현된다.

① V, E, ε는 일반적인 공격 트리에서의 정의와 같다.

② Label은 각 앳지와 연관된 공격 명칭을 의미한다.

③ $SIG_{u, v}$는 공격 신호를 의미한다.

정의 4. Incident-Choice는 공격 트리에서 상태전이에 변화를 줄 수 있는 사건들과 관련된 사건들의 그룹으로 정의한다.

정의 5. $SIG_{u, v}$는 Atomic attack을 구성하는 Incident 순서$(incident_{i,1}, incident_{j,2}, ..., incident_{m,n})$를 위한 Incident-Choice의 순서$<incident\text{-}choice_1, incident\text{-}choice_2, ... , incident\text{-}choice_n>$를 의미한다.

Ⅲ. 확장된 사이버 공격 트리

본 논문에서 제안한 사이버 공격 트리는 일반적인 공격 트리를 확장한 사이버 공격 트리(E-CAT: Extended-Cyber Attack Tree, 이하 E-CAT)이다. E-CAT는 서브노드들이 그들의 서브목적을 달성하기 위하여 수행하는 공격 방법을 강화하고 새롭고 변형된 공격방법을 표현할 수 있도록 조합식에 'CON' 조합을 추가시켰다. 그리고 Boolean 표현식을 사용함으로써 공격 시나리오를 간결하게 표현하고 향후 사이버 공격 모델을 구현할 때 공격 에이전트에 공격 시나리오 저장 공간을 줄이고 공격 통제 모듈과 공격 에이전트간 공격 시나리오 전송량을 감소시킬 수 있다. 또한 상위 노드로 이동하기 위하여 하위 노드들의 공격 절차를 선택할 때 가장 공격 성공 가능성이 높은 공격 경로를 선택할 수 있게끔 AGP 요소를 추가하였다. 확장된 사이버 공격 트리(E-CAT)의 형식은 다음과 같다.

E-CAT=<N, R, ε, B, AGP>

정의 1. *N(Node)*: 공격 트리를 구성하는 노드들의 집합. N은 leaf_node들과 internal_node들로 구성된다. $n \in N$은 루트노드이며, 공격의 최종목표를 의미한다.

i) $\{leaf\_nodes, internal\_nodes\} \in N$, $n \in internal\_nodes$

ii) $leaf\_nodes \cup internal\_nodes = N$

iii) $leaf\_nodes \cap internal\_nodes = \varnothing$

정의 2. *R(Route)*: 공격 트리에서 각 노드사이를 연결시켜 주는 공격상태들의 집합. $R \subseteq N \times N$로 표현한다. $r(n_1, n_{1.1}) \in R$은 자식노드(Child node) $n_{1.1}$에서 부모노드(Parent node) n_1까지 공격상태 전이를 나타낸다 $(n_1, n_{1.1} \in N)$.

정의 3. *ε: <n, composition>* 형식의 튜플들의 집합.

i) $n \in internal\_nodes$, $composition \in \{AND, OR, CON\}$

ii) *AND-composition: if* r_1 *and* r_2 *are connected by AND-composition, do all* r_1 *and* r_2

iii) OR-composition: if r_1 and r_2 are connected by OR-composition, do either r_1 or r_2

iv) CON-composition: if r_1 and r_2 are connected by CON-composition, do r_1 and decide whether do r_2 or not according to attack condition or target's environments

정의 3에서 CON은 공격 대상 네트워크 시스템의 환경과 공격 조건에 따라 공격자가 기존에 표현이 어려웠던 새롭게 변형된 공격방법을 표현하고 공격을 보완할 목적으로 공격절차를 추가하는 것이다. 예를 들어, 09년도 7.7 DDoS 대란때 기존의 디도스 공격에 없었던 악성코드 wversion. exe을 이용하여 하드디스크의 물리적인 첫 시작 위치에 'Memory of the Independence Day'라는 문구를 삽입하여 소프트웨어적 하드디스크 손상시키는 공격방법이 새롭게 발견되었다. 이 공격방법을 추가시키기 위해서는 기존의 AND와 OR 조합대신 CON 조합을 추가함으로써 능동적으로 공격방법을 표현할 수 있게 한다. 또한, 기존의 공격방법으로는 성공률이 낮을 때, CON 조합으로 공격을 보강할 수 있는 또 다른 공격방법을 추가하여 공격 성공률을 증가시킨다. 최근에 사이버 공격방법이 여러 가지의 방법을 혼합하여 사용하기 때문에 공격절차가 매우 복잡하다. 그래서 기존의 공격 트리는 고도화된 공격방법을 표현할 수 없었으나, E-CAT는 CON 조합을 활용하여 정교한 공격방법도 표현할 수 있다.

정의 4. *B(Boolean Expression)*: 최하위 노드부터 최상의 노드까지의 공격진행, 즉, 조합의 순서를 기호로 표시한다. *AND* → ∧, *OR* → ∨, *CON* → ⊻로 대신한다.

예를 들어 $A_{DoS\ Attack}$=*(UDP Flooding OR SYN Flooding OR ICMP Flooding)*은 기호와 조합을 사용하여 $B_{DoS\ Attack}=(n_1 \vee n_2 \vee n_3)$처럼 간단하게 표현할 수 있다. 또한, 이 공격이 성공하려면 n_1, n_2, n_3가 순차적으로 조합에 맞춰서 발생하여야 한다. 만약, 공격 제한시간에 n_1, n_2, n_3가 같은 순서대로 발생하지 않거나 조합 기호가 다르게 나타나면 공격을 성공할 수 없다. 또한, 사이버 공격 모델이 E-CAT를 사용할 경우 데이터베이스에 원본의 공격 시나리오를 저장해 두고 공격 에이전트에 공격 시나리오를 탑재할 때 Boolean 표현식 형태로 추출함으로써 공격 시나리오가 노출되어도 정확한 공격방법을 알아낼 수 없으며, 공격 통제 모듈과 공격 에이전트간 공격 시나리오를 송수신할 때 전송량을 감소시킨다.

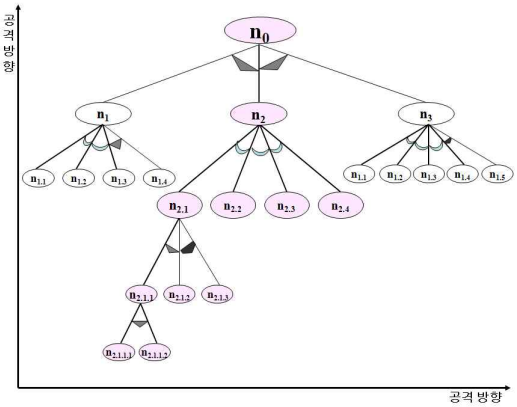
정의 5. *AGP(Attack Generation Probability)*: 어떤 서브노드의 공격목적을 달성하기 위해서 서브목적과 관련된 모든 공격행위 대비 자식노드의 모든 공격발생 비율을 의미한다[14]. 예를 들어, 어떤 공격 트리의 총 노드가 $AT_N=\{N_1, N_2, N_3..., N_n\}$이면서 자식노드가 N_2만 있어서 자식노드가 $AT_{N2}=\{n_1, n_2, n_3\}$일 경우 N_2의 AGP은 N_2의 노드 수인 3을 총 노드수 n으로 나눈 값이 되는 것이다. 즉, 3/n이 된다. AGP는 부모노드가 그의 공격목적을 달성하기 위해 자식노드를 생성하는 확률로 볼 수 있다. 자식노드들이 증가하면 증가할수록 공격 제한시간에 모든 공격을 실행하여 공격목적을 달성할 수 있는 확률이 낮아진다. 즉, 공격목적을 달성하기 여러 가지 공격방법을 혼합하여 사용하면 공격이 성공할 확률이 높아질지라도, 그만큼 공격대상 시스템에 탐지될 가능성도 높아진다. 최종 공격목적을 달성하기 위해 최종노드 다음의 서브노드를 선택할 때 AGP 값이 낮은 것을 선택한다. 또한, 공격이 진행될 때 AGP 값을 확인하여 공격경로 상태 단계를 확인할 수 있다. AGP 값이 높으면 그 만큼 상위노드의 공격 경로에 있는 것이며 최종 공격목적에 가까워졌다는 것을 의미한다.

Ⅳ. 확장된 사이버 공격 트리 평가

확장된 사이버 공격 트리(E-CAT)는 기존의 공격 트리가 공격방법이나 절차를 표현하는데 갖고 있던 한계점을 보안해서 제안된 공격 트리이다. 우선 조합식에 CON 조합을 추가함으로써 공격자의 결정이나 공격목표의 환경을 고려하여 단순한 공격절차뿐 만 아니라 새롭게 변형

되고 복잡한 공격방법도 표현할 수 있게 하였다. 또한, 기존의 공격 트리에서는 공격절차와 방법이 데이터베이스에 시나리오 형태로 저장되어 시스템의 저장공간을 부족하게 하거나 공격 에이전트가 공격을 변경할 때 새로운 공격경로를 전송할 때 서술식으로 전송하여 정보 노출의 위험이 있었다. 그러나 E-CAT는 Boolean 표현식으로 공격방법과 절차를 저장하여 메모리를 절약하고 시스템과 에이전트간 전송량을 감소시킨다. 마지막으로 AGP을 적용하여 공격 경로를 공격 트리에서 독립적으로 선택할 수 있게 하였다.

본 장에서는 DoS 공격[15]을 예로 들어 이전 장에서 제시한 확장된 사이버 공격 트리를 평가한다. DoS 공격의 시나리오는 표 1과 같다. 공격 시나리오에서 Goal은 공격목적을 나타내고 MPI(Main Point of Impact)는 주 공격 취약점이며, Pre-condition은 공격 사전조건을 의미한다. Attack은 공격방법이고 Post-condition은 공격이후의 공격대상 시스템의 상태를 나타낸다. 본 논문에서는 DoS 공격인 'UDP Flooding, SYN Flooding, ICMP Flooding' 공격 중에 'SYN Flooding' 공격을 중심으로 공격 트리를 구성할 것이며, 사이버 공격 트리를 평가하는 데 사용된다. 표 1의 공격 시나리오를 공격 트리로 표현한다면 그림 1과 같다.



<그림 1> DoS 공격 트리

표 1과 그림 1을 확장된 사이버 공격 트리 형식으로 표현한다면 다음과 같다.

$E\text{-}CAT_{DoS\ Attack} = \langle N, R, C, B, AGP \rangle$

① $N = \{n_1, n_2, n_3, n_{1.1}, n_{1.2}, n_{1.3}, n_{1.4}, n_{2.1}, n_{2.2}, n_{2.3}, n_{2.4}, \ldots, n_{1.2.3.\ x}\}$로 표현할 수 있다. 본 논문에서는 n_2 = *SYN*

<표 1> DoS 공격 시나리오

1. Denial of Service Attack
   - Goal: Take target server out of action for a few hours
   - MPI: Server
   - Pre-condition: the weak security countermeasures and spoofing target server's IP address
   - Attack
     OR-Comp. 1. UDP Flooding, 2. SYN Flooding, 3. ICMP Flooding
   - Post-condition: the service of target server is interrupted for a few hours

   1.2 SYN Flooding attack
   - Goal: Exploit TCP Three-way handshaking vulnerability to perform overflow Backlog queue
   - MPI: Three Way Hand Shaking vulnerability in TCP
   - Pre-condition: Attacker executes a hacking program on target system for spoof server's IP address
   - Attack
     AND-Comp. 1. Identify source IP address which is unreachable
     OR-Comp. 1. Execute automated hacking program
     OR-Comp. 1. use "whois"system, 2. execute "port scan program"
     2. Use Social engineering method using insider
     CON-Comp. 3. Use "trusted insider"
     2. Send SYN packet to target server for connection
     3. Don't send ACK packet after received SYN/ACK packet from target server
     4. Continue to send a fake connection to target server in the "Half Open" state until backlog queue is full
   - Post-condition: Denial of all connection request in this port

*Flooding*로 한정하여 설명한다. n_2 = {$n_{2.1}$, $n_{2.2}$, $n_{2.3}$, $n_{2.4}$, $n_{2.1.1}$, $n_{2.1.2}$, $n_{2.1.3}$, $n_{2.1.1.1}$, $n_{2.1.1.2}$}

② R={$r(n_{2.1.1.1})$, $r(n_{2.1.1.2})$, $r(n_{2.1.1})$, $r(n_{2.1.2})$, $r(n_{2.1.3})$, $r(n_{2.1})$, $r(n_{2.2})$, $r(n_{2.3})$, $r(n_{2.4})$, $r(n_2)$} 여기서, $r(n_{2.1.1})$은 $r(n_{2.1.1.1}, n_{2.1.1.2})$과 같다. 따라서 $r(n_{2.1})$=$r(n_{2.1.1}, n_{2.1.2}, n_{2.1.3})$로 표현할 수 있다.

③ ε={<($n_{2.1.1.1}$, $n_{2.1.1.2}$), *OR*>, <($n_{2.1.1}$, $n_{2.1.2}$), *OR*>, <(($n_{2.1.1}$, $n_{2.1.2}$), $n_{2.1.3}$), *CON*>, <($n_{2.1}$, $n_{2.2}$), *AND*>, <($n_{2.2}$, $n_{2.3}$), *AND*>, <($n_{2.3}$, $n_{2.4}$), *AND*>} 공격 상태들의 집합에서 CON 조합을 추가하여 복잡한 공격방법을 표현하고 공격목적을 달성할 수 있도록 공격방법을 보강하였다. 공격방법을 보강하는 목적으로 예를 들어 설명하면 다음과 같다. $n_{2.1}$의 공격목적이 공격대상 시스템의 IP 주소를 알아내는 것으로 서브공격 상태는 일반적으로 $\varepsilon n_{2.1}$=<($n_{2.1.1}$, $n_{2.1.2}$), *OR*>으로 종료된다. 즉, 해킹 프로그램을 실행시키거나 사회공학적 방법을 통하여 알아내는 것으로 마무리된다. 그러나 공격대상 시스템에 대한 정보를 일체 제공하고 있지 않고 시스템 접근을 차단하는 방화벽이 설치되어 해킹 프로그램을 사용해도 IP 정보를 알아낼 수 없고 물리적으로 공격대상 시스템에 접근할 수 없어서 사회공학적 방법을 이용할 수 없다면 서브 공격목적을 달성할 수 없다. 이럴 경우에 CON(믿음직한 내부자를 이용)으로 연결된 <(($n_{2.1.1}$, $n_{2.1.2}$), $n_{2.1.3}$), *CON*>을 추가함으로써 $\varepsilon n_{2.1}$=<($n_{2.1.1}$, $n_{2.1.2}$), *OR*>가 표현할 수 없는 공격방법에 유연성을 줄 수 있으며, 실패할 확률도 감소시킨다. 다시 말해서 $\varepsilon n_{2.1}$= <(($n_{2.1.1}$, $n_{2.1.2}$), *OR*), $n_{2.1.3}$), *CON*>를 이용하여 새로운 공격방법이나 이전에 표현하지 못했던 공격방법을 묘사할 수 있고, 공격측면에서는 공격방법을 보강한 후 성공확률을 높이게 할 수 있다.

④ Boolean Expression은 최말단 노드를 자식으로 갖는 노드부터 공격목적이 있는 모든 서브노드들의 공격절차를 간략하게 표현할 수 있다. 예를 들어, n_2의 공격경로를 서술식으로 표현한다면 Attack Path=<(((*Use "Whois" System OR Execute "Port Scan" Program) OR Social engineering method using insider) CON Use trusted insider) AND Send SYN packet to target server for connection AND Don't send ACK packet after received SYN/ACK packet from target server AND Continue to send a fake connection to target server in the "Half Open" state until backlog queue is full*>와 같다. 반면에 Boolean 표현식은 표 2처럼 $E\text{-}CAT_{n2}$=((($n_{2.1.1.1}$ ∨ $n_{2.1.1.2}$) ∨ $n_{2.1.2}$ ⩒ $n_{2.1.3}$) ∧ $n_{2.2}$, ∧ $n_{2.3}$, ∧ $n_{2.4}$)와 같이 간단하게 표현할 수 있다. Boolean 표현식으로 공격 경로를 표현하는 것은 서술식으로 표현하는 것보다 몇 가지의 장점을 갖는다. 첫째, 공격 경로를 저장하는 D/B의 공간을 줄일 수 있고, 둘째, 사이버 공격 모델에 적용될 경우 공격 통제 모듈과 공격 에이전트간 공격 경로를 송수신할 때 데이터 전송량을 감소시킬 수 있다. 마지막으로, 공격대상 시스템에게 공격이 탐지되더라도 정확한 공격방법을 노출시키지 않는다.

<표 2> SYN Flooding 공격의 Boolean 표현식

Subnode	Boolean Expression
n2.1.1	n2.1.1.1∨n2.1.1.2
n2.1	(n2.1.1.1∨n2.1.1.2)∨n2.1.2⩒n2.1.3
n2	((n2.1.1.1∨n2.1.1.2)∨n2.1.2⩒n2.1.3)∧n2..2,∧n2.3,∧n2.4

또한, Andrew P. Moore의 "Attack Modeling for Information Security and Survivability'[16]에서 공격절차를 나타내는데 사용된 공격 시퀀스(Attack Sequence)보다 더 정교하다. 표 1의 공격 시나리오를 공격 시퀀스로 표현한다면 다음 표 3과 같다.

<표 3> SYN Flooding 공격의 공격 시퀀스

Subnode	Boolean Expression
n2.1.1	n2.1.1.1, n2.1.1.2
n2.1	(n2.1.1.1, n2.1.1.2), n2.1.2, n2.1.3
n2	((n2.1.1.1, n2.1.1.2), n2.1.2, n2.1.3), n2..2, n2.3, n2.4

위 표에서 보시는 바와 같이 공격 순서는 기호를 통해

서 알 수 있으나 각 공격 노드들간에 공격 형태를 알 수 없다. 각 노드별로 공격을 모두 실행하는 것인지, 아니면 그 중에 하나만 수행하면 되는 것인지 알 수 없다.

⑤ AGP는 OR 조합으로 연결되어 있는 공격방법을 선택할 때 사용된다. OR 조합으로 연결되어 있는 부모 노드가 자식 노드를 가지고 있을 때 자식 노드의 숫자를 총 노드 숫자로 나누어 AGP 값을 계산한다. 예를 들어 디도스 공격 목적을 달성하기 위해 공격방법을 선택한다고 가정하자. SYN Flooding의 공격 트리에서 달성해야 할 서브목적은 총 3가지이며, 조합으로 연결되어 있는 노드는 총 7개이다. $n_{2.1.1}$의 AGP는 총 7개의 노드 중에서 2개의 노드의 공격을 수행해야 하므로 2/7(0.29)이다. 각 최종 공격목적을 달성하기 위한 서브목적의 AGP는 $AGP_{n2.1}$=0.57 $AGP_{n2.}$ =1이다. AGP는 공격방법을 선택할 때 사용된다. 예를 들어 SYN Flooding의 AGP는 서브목적의 AGP를 모두 합한 값이므로 1.86이 된다. 만약 ICMP Flooding의 AGP의 값이 2.5라면 공격 에이전트는 AGP 값이 낮은 SYN Flooding을 선택하게 된다. 그 이유는 AGP의 값이 높으면 실행해야 할 노드들과 달성해야 할 서브목적들이 많다는 의미가 되며, 공격대상 시스템에게 탐지될 확률도 높아지게 된다. 일반적인 공격 트리와 E-CAT를 종합적으로 비교하면 다음 표 4와 같다.

<표 4> 일반적인 공격 트리와 E-CAT 비교

대 상	GAT	E-CAT
구성 요소	ν, ε, θ	N, R, ϵ, B, AGP
공격 조합	AND, OR	AND, OR, CON
공격 우선순위	확실치 않음	AGP 값에 따라 결정
공격 표현	시나리오식	Boolean 표현식
조합 기호	없음	∧, ∨, ⊻
트리 표현식	시나리오, 조합식 구성	기호와 부호로 구성

E-CAT이 일반적인 공격 트리보다 구성요소가 2개가 추가되어 복잡하게 보일 수 있으나, 공격트리를 표현하는데 있어서는 좀 더 구체적이고 명확하게 표현할 수 있다. 공격조합에 CON 조합을 추가함으로써 기존의 AND와 OR 조합으로 표현할 수 없었던 새로운 변형된 공격방법을 표현할 수 있으며, AGP는 E-CAT이 공격방법을 독립적으로 선택할 수 있게 하고 공격 전이 상태를 파악할 수 있게 한다. 공격 표현방법은 일반적인 공격 트리에서 공격노드별 공격방법을 모두 서술식으로 나열한 것을 노드별 Boolean 표현식을 활용하여 간단하게 표현할 수 있게 하였다. 기호와 부호로 표현된 공격 트리가 사이버 공격 모델에 적용할 경우 공격 에이전트에 공격방법과 절차를 전송하거나 저장할 경우, 저장공간을 줄일 수 있고 공격 에이전트에 송신하는 공격 시나리오 전송량을 감소시킬 수 있으며 공격대상 시스템에 공격방법이 탐지되더라도 공격에 대한 정보를 노출시키지 않는다.

Ⅳ. 결론

본 논문은 사이버 공격 모델 설계를 위해 확장된 공격 트리를 제안하였다. *E-CAT=<N, R, ϵ, B, AGP>*은 기존의 *Attack Tree = (ν, ε, θ)*에서 공격방법이나 절차를 표현하는데 갖고 있던 한계점을 보완해서 제안된 공격 트리이다. 기존의 공격 트리에서 *= (ν, ε, θ)*에서 CON 조합 속성, Boolean 표현식(B)과 AGP을 추가하여 공격 표현의 다양성, 공격 성공 확률, 공격 시나리오의 표현 축약, 공격방법 선택의 효과성을 반영하였다.

조합식 'θ' 요소인 AND, OR 조합에 CON을 추가하여 공격조건이나 목표 환경을 고려하여 단순한 공격절차뿐 만 아니라 정교하고 복잡한 공격절차도 표현하고 공격 방법을 보강할 수 있게 하였다. Boolean 표현식(B)은 공격상태를 서술식 표현에서 기호와 부호로 표현함으로써 공격상태와 노드, 목적 등을 간단하게 표현할 수 있고 사이버 공격 모델에 적용할 경우 서버와 공격 에이전트 간의 공격 시나리오 전송량을 감소시킨다. AGP은 공격 생성 확률로 공격대상 시스템에 탐지되지 않고 최종 공

격목적을 달성할 수 있도록 공격방법을 최소의 공격경로를 선택할 수 있도록 하였다.

향후에 E-CAT을 적용한 사이버 공격 모델을 설계 및 구현하여 네트워크 취약성 분석에 시뮬레이션을 수행할 계획이다.

참고문헌

[1] Ariel Futoransky et al, "Building computer network attacks," Technical report, Core Labs, Core Security Technology, 2003.

[2] 엄정호 외 2명, "보안 안전성을 위한 자동화 보안진단평가 시스템에 관한 연구," 디지털산업정보학회 논문지, 제5권 제4호, December, 2009, pp. 109-116.

[3] 엄정호 외 3명, "*사이버 공격과 보안 기술*," 홍릉과학출판사, 2009, pp. 3-9.

[4] Curt A. Carver, et al, "Military Academy Attack/Defense Network," Annual IEEE Information Assurance Workshop, Jun, 2002, pp. 29-34.

[5] Kristopher Daley, Ryan Larson, and Jerald Dawkins, "A Structural Framework for Modeling Multi-stage Network Attacks," Proceeding of the International Conference on Parallel Processing Workshops, Aug, 2002, pp. 5-10.

[6] Bruce Schneier, "Attack Trees: Modeling Security Threats," Dr. Dobb's Journal, Dec, 1999.

[7] Cynthia Phillips and Laura Painton Swiler, "A graph-based system for network vulnerability analysis," Proceedings of the 1998 workshop on new security paradigms, ACM press, Sept, 1998, pp. 71-79.

[8] Wei Wang and Thomas E. Daniels, "A Graph Based Approach Toward Network Forensics Analysis," ACM Transactions on Information and Systems Security, Vol. 12, No. 1, Oct, 2008, pp. 401-433.

[9] Bruce Schneier, "*Secrets and Lies: Digital Security in a Networked World*," John Wiley & Sons, 2000.

[10] Vineet Saini, et al, "Threat modeling using attack trees," Journal of Computing Sciences in Colleges, Vol. 23, Issue4, Apr, 2008, pp. 124-131.

[11] Nayot Poolsapassit and Indrajit Ray, "Investigating Computer Attacks using Attack Trees," Advances in Digital Forensics Ⅲ, Vol. 242, Nov, 2007, pp. 331-343.

[12] Seyit Anmet Camtepe and Bulent Yener, "Modeling and Detection of Complex Attacks," Proceedings of the third international conference on security and privacy in communication networks, Sept, 2007, pp. 234-243.

[13] Hubert Comon, et al, "*Tree Automata Techniques and Applications*," TATA, Sept, 2002.

[14] Jung ho Eom, et al, "Active Cyber Attack model for Network system's Vulnerability Assessment," International Conference on Information Science and Security(ICISS 2008), Jan, 2008, pp. 153-158.

[15] Jong-yeub Lee, et al, "Monitoring and Investigation of DoS Attack," KNOM Reveiw, Vol. 6, No. 2, Feb, 2004, pp. 33-40.

[16] Andrew P. Moore et al, "Attack Modeling for Information Security and Survivability," Technical Note, CMU/SEI-2001-TN-001, Mar, 2001.

■ 저자소개 ■

엄 정 호
Eom, Jung Ho

2010년~현재 성균관대학교 정보통신공학부 BK21 연구교수
2010년 대한민국 공군 장교
2008년 성균관대학교 컴퓨터공학과(박사)
2003년 성균관대학교 컴퓨터공학과(석사)
1994년 공군사관학교 항공공학과(학사)

관심분야 : 사이버전, 사이버공격모델, 접근제어, 위험분석
E-mail : eomhun@gmail.com

박 선 호
Park, Seon Ho

2007년~현재 성균관대 전자전기컴퓨터공학과 박사과정
2007년 성균관대학교 컴퓨터공학 석사
2005년 성균관대학교 정보통신공학부 학사

관심분야 : 유비쿼터스 컴퓨팅, 시스템 보안, 네트워크 보안, 접근제어 모델 및 검증 방법론
E-mail : shpark@imtl.skku.ac.kr

정 태 명
Chung, Tai M.

1995년~현재 성균관대학교 컴퓨터공학과 교수
1995년 Purdue University W. Lafayette, IN, U. S. A. 컴퓨터공학 졸업(박사)
1987년 University of Illinois Chicago IL, U. S. A. 컴퓨터공학과 졸업(석사)
1984년 University of Illinois Chicago IL, U. S. A. 전자계산학과 졸업(학사)
1981년 연세대학교 전기공학과 졸업(학사)

관심분야 : 통합보안관리, 네트워크, 무선망
E-mail : tmchung@ece.skku.ac.kr

논문접수일 : 2010년 8월 7일
수 정 일 : 2010년 8월 15일(1차), 8월 23일(2차)
게재확정일 : 2010년 8월 28일