情報保護學會論文誌
第 21 卷 第 3 號, 2011. 6

능동적인 사이버 공격 트리 설계: 애트리뷰트 접근

엄 정 호†
대전대학교

An Architecture of a Dynamic Cyber Attack Tree: Attributes Approach

Jung-ho Eom†
Daejeon University

요 약

본 논문에서는 복잡하고 변형된 공격을 탐지할 수 있는 사이버 공격 모델을 위해서 공격 시나리오를 유연하게 표현할 수 있는 능동적인 사이버 공격 트리를 제안하였다. 공격 트리는 네트워크 시스템에 대한 다양한 공격에 대해 보안 대책을 수립할 수 있도록 체계적이고 조직적인 해결책을 제시한다. 기존의 공격 트리는 정점(ν), 앳지(ε) 그리고 조합(θ)을 이용하여 공격 시나리오를 표현할 수 있다. 그러나 제한적인 애트리뷰트를 인해서 복잡하고 변형된 공격을 표현하기에는 한계가 있다. 우리는 공격 트리 애트리뷰트에 조합 요소와 위협발생확률을 추가하여 기존에 공격 트리가 갖고 있던 한계점을 해결하였다. 우선 조합요소를 강화하여 공격 순서의 모호성을 줄였으며 변형된 공격 표현의 유연성을 향상시켰다. 위협발생확률을 추가하여 자식노드부터 부모노드까지의 공격단계별 위험수준을 알 수 있게 하였다.

ABSTRACT

In this paper, we presented a dynamic cyber attack tree which can describe an attack scenario flexibly for an active cyber attack model could be detected complex and transformed attack method. An attack tree provides a formal and methodical route of describing the security safeguard on varying attacks against network system. The existent attack tree can describe attack scenario as using vertex, edge and composition. But an attack tree has the limitations to express complex and new attack due to the restriction of attack tree's attributes. We solved the limitations of the existent attack tree as adding an threat occurrence probability and 2 components of composition in the attributes. Firstly, we improved the flexibility to describe complex and transformed attack method, and reduced the ambiguity of attack sequence, as reinforcing composition. And we can identify the risk level of attack at each attack phase from child node to parent node as adding an threat occurrence probability.

Keywords: IM-ACM, Insider threat, Access Control

I. 서 론

최근 '스턱스넷(Stuxnet)'이라는 신종 웜 바이러스가 이란 원자력발전소를 공격한 후 중국의 산업시설 시스템까지 감염시킴으로써 보안업계에서는 대책 마련에 고심하고 있다. 스턱스넷은 일반 개인용 컴퓨터를 공격하는 기존 웜과는 달리 독일 지멘스가 생산한 소프트웨어만 공격하여 파괴시키는 것이 특징이다. 특정 시스템의 취약점을 이용한 사이버 공격 전술이 고도화됨에 따라 취약점을 식별하고 공격에 대한 피해를 최소화하기 위해서는 최대한 빠른 시간 내에 침입을 탐지해야 한다. 침입 탐지에 사용되는 메커니즘 중에는 공격 프로세스를 트리 형태로 표현한 공격 트리(Attack Tree) 모델이 있다[1,2].

공격 트리는 네트워크 시스템에 대한 다양하고 복잡한 공격에 대해 보안 대책을 수립할 수 있도록 규칙적

접수일(2010년 10월 25일), 수정일(2011년 1월 3일),
게재확정일(2011년 3월 15일)
† 주저자, eomhun@gmail.com

이고 체계적인 방법을 갖는다[3,4]. Bruce Schneier는 그의 논문[5]에서 공격 트리는 "각 노드들의 자식 노드('AND' 와 'OR' 노드)들이 실행이 완료되면 자식 노드의 목적을 달성할 수 있고 모든 자식 노드들의 실행이 완료되어 최종적으로 루트(최종) 노드가 실행되면 최종 공격목적을 달성하게 된다"고 밝혔다. 그러나 기존의 공격 트리 모델이 AND와 OR 조합으로만 구성하고 있기 때문에 복잡하고 변형된 공격 방법을 표현하는데 한계가 있다. 또한, 공격의 우선순위를 포함하고 있지 않다. 본 논문은 기존 공격 트리의 제한사항을 애트리뷰트 관점에서 해결하고 능동적인 사이버 공격 트리를 설계하였다.

본 논문은 2장에서 관련 연구와 기존 공격 트리에 대한 제한사항을 명시하고 3장에서 해결책을 제시한다. 4장에서는 능동적인 사이버 공격 트리를 제안하고 평가한다. 마지막으로 5장에서 결론을 맺는다.

II. 관련연구

2.1 공격 트리(Attack Tree)

공격 트리는 네트워크 시스템을 대상으로 한 다양한 공격에 대해 보안 대책을 수립할 수 있도록 논리적이고 체계적인 방법을 제공한다. 공격 트리 구조는 루트 노드(Root Node)의 최종 목적, 최하위 노드(Leaf Node)들과 중간 노드(Internal Node)들의 목적을 달성하기 위한 다양한 공격 방법들로 묘사된다. 노드들은 일반적으로 선택 가능한 'OR' 노드와 목적을 달성하기 위해 반드시 수행해야 하는 'AND' 노드로 구성된다[6-8]. 본 논문에서는 Bruce Schneier가 제시한 'Attack Tree'[5]를 살펴본다.

공격 트리는 정점(ν), 앳지(ε), 조합(θ)으로 이루어진다. 정점(ν)는 공격이나 공격 목적을 나타내는 노드들의 집합이고, 앳지(ε)는 자식 노드에서 공격을 수행하여 부모 노드로 이동하는 공격 전이 상태들의 집합을 의미한다.

공격 트리 표현식은 Attack Tree = (ν, ε, θ)와 같으며, 각 구성요소에 대한 정의는 아래와 같다.

① V는 하위 공격이나 목적을 나타내는 노드들의 집합. $\nu \in V$는 공격자의 최종 목적을 나타내는 트리의 최상위 루트 노드를 나타낸다. V는 leaf_node와 internal_node로 구분된다. leaf_node는 [그림 1]에서 v4처럼 최하위 노드로 다른 자식 노드들과 조합으로 연결되어 있지 않으며 internal_node는 v1과 같이 중간 노드로 다른 노드들과 조합으로 연결되어 있다.

(i) leaf_nodes ∪ internal_nodes = V; leaf_node와 internal_node들의 목적을 달성해야만 최종 목적을 달성할 수 있다.

(ii) leaf_nodes ∩ internal_nodes = ∅; leaf_node이면서 internal_node가 될 수 없다.

(iii) $\nu \in$ internal_node; ν는 internal_node의 요소이다.

② $\varepsilon \subseteq V \times V$ 는 공격 트리에서 앳지(edge)의 집합. edge(u, v) $\in \varepsilon$는 자식 노드 v에서 부모 노드 u까지 상태 전이를 나타내는 단위 공격으로 정의한다(u, v $\in$ V).

③ ε는 $\langle \nu$, composition$\rangle$ 형식인 튜플들의 집합.

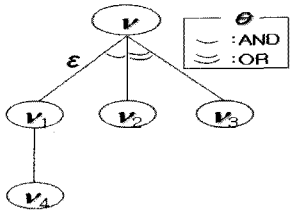
(i) $\nu \in$ internal_nodes and; 조합식으로 연결될 수 있는 노드는 internal_node이다.

(ii) composition = [AND, OR]; 조합식은 'AND, OR'로 구성된다.

(iii) node $\nu \in$ internal_nodes이면서 노드에 연결된 모든 앳지가 AND에 의해 연결되어 있으면 AND-composition(이하 AND)으로 정의하며, AND로 연결되어 있는 모든 internal_node는 반드시 공격 이벤트를 수행해야 한다. 즉, [그림 1]에서 AND로 연결된 v1과 v2는 모두 공격 이벤트를 수행해야 한다.

(iv) node $\nu \in$ internal_nodes이면서 노드에 연결된 모든 앳지가 OR에 의해 연결되어 있으면 OR-composition(이하 OR)으로 정의하며, OR로 연결되어 있는 모든 internal_node들 중에 하나의 노드만 공격 이벤트를 수행하면 된다.

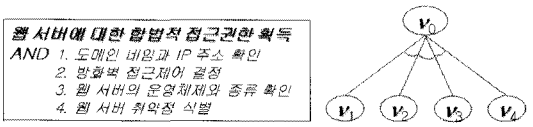
Bruce Schneier의 Attack Tree는 [그림 1]과 같이 표현할 수 있다.



[그림 1] Attack Tree

2.2 기존 공격 트리의 제한사항

기존의 공격 트리는 복잡하고 변형된 공격을 묘사하는 데 한계가 있다. 즉, 공격방법의 순서, 우선순위, 위험수준 등의 정보를 포함하지 않기 때문에 공격 이벤트의 발생순서나 단계별 진행에 따라 위험수준을 알 수 없다. 예를 들어, 반드시 노드들의 공격 이벤트가 차례대로 발생해야 침입으로 판단할 수 있는 공격을 기존의 공격 트리에서는 순서에 상관없이 공격 이벤트들이 공격 트리의 노드들에 포함되어 있기 때문에 침입이라고 판단하여 'False Positive'가 증가할 수 있다. 본 장에서는 기존의 공격 트리의 제약사항을 애트리뷰트 관점에서 식별한다.

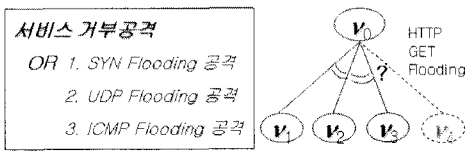


〔그림 2〕 웹 서버 권한 획득 공격 시나리오 및 트리

첫째, 공격 노드들을 구성하는 조합 'composition = {AND, OR}'에는 공격 이벤트 발생 순서를 정의하고 있지 않다. 즉, 〔그림 1〕에서 AND로 구성된 공격 노드 v1과 v2 중에 어떤 노드부터 공격 이벤트가 발생해야 하는지 모른다. 예를 들어 '웹서버 합법적 접근권한 획득'이라는 공격을 수행할 때 공격 트리를 〔그림 2〕와 같이 구성할 수 있다. 그리고 AND 조합으로 구성된 첫 번째 '도메인 네임과 IP 주소 확인' 이벤트부터 마지막 '웹 서버 취약점 식별'까지 순서대로 발생해야 최종 공격목적을 달성할 수 있다. 공격 트리에서 보면 v1부터 v4까지 순차적으로 공격 노드의 이벤트가 발생하지 않으면 최종 공격목적을 달성할 수 없다. 공격 트리에서 이벤트 발생 순서를 정하지 않았을 경우, $\forall(\nu_1\wedge\nu_2\wedge\nu_3\wedge\nu_4)\rightarrow\nu_0$나 $\forall(\nu_2\wedge\nu_3\wedge\nu_1\wedge\nu_4)\rightarrow\nu_0$ 같다는 것을 의미한다. 만약, 공격 트리를 적용한 침입탐지 메커니즘에서 순서에 상관없이 4개의 하부 노드의 공격 이벤트가 발생하면 탐지모듈은 이벤트를 침입이라고 판단하여 False Positive rate가 증가한다. 〔그림 2〕의 공격 시나리오에서는 웹 서버의 운영체제와 종류를 식별하지 않고서 웹 서버의 취약점을 파악할 수는 없다.

둘째, OR로 구성된 노드들이나 AND와 OR로 구성된 노드들일 경우, 어떤 노드를 먼저 선택해야 할지 우선순위를 알 수 없다. 즉, 〔그림 1〕에서 $\exists(\nu_1\wedge\nu_2\vee\nu_3)\rightarrow\nu_0$은 AND 조합으로 구성된 v1과 v2를 선택해야 할지, v3을 선택해야 할지 정해진 규칙이 없다. 우선순위는 공격대상 시스템 환경과 설치된 보안대책 등을 고려하여 명확히 결정해야 한다.

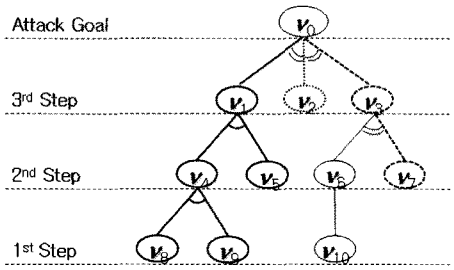
셋째, 복잡하고 변형된 공격 방법을 표현하기에는 AND나 OR 조합으로는 충분하지 못하다. 예를 들어, 분산서비스거부공격[9]에서 일반적으로 SYN Flooding 공격, UDP Flooding 공격, ICMP Flooding 공격에 의해 발생한다. 〔그림 3〕에서 서비스거부공격을 성공하기 위해서는 OR 조합으로 연결된 첫 번째 SYN Flooding 공격노드부터 마지막 ICMP Flooding 공격노드 중에 단 하나만 성공시키면 된다. 그러나 〔그림 3〕처럼 세 가지의 공격방법으로 공격 트리를 구성한다면 작년 7.7 DDoS 대란 공격 방법 중에 HTTP GET을 계속적으로 요청하여 웹 서버를 마비시키는 HTTP GET Flooding 공격은 탐지할 수 없다. 또한, DDoS 공격 마지막 단계에서는 처음으로 악성코드 wversion.exe(2nd)(Win-Trojan/Destroyer. 37264)가 하드디스크의 물리적인 첫 시작 위치에 'Memory of the Independence Day'라는 문구를 이용하여 소프트웨어적으로 하드디스크를 손상시키는 결과도 예상하지 못한다[10].



〔그림 3〕 서비스거부공격 시나리오 및 트리

마지막으로, 최하위 노드에서 최종 노드까지 공격 이벤트가 진행되면서 공격에 대한 위협수준을 평가할 수 있는 방법이 없다. 공격 트리에서 최하위 노드에서 공격 이벤트가 발생하여 진행됨에 따라 공격 단계별, 위협수준을 알 수 없다. 〔그림 4〕에서 v1을 선택할 경우에는 3단계(1~3단계) 과정에서 5개의 노드(v1, v4, v5, v8, v9)들의 공격 이벤트들이 발생한다. v3을 선택할 경우에는 공격대상 시스템 환경에 따라 양자 선택할 수 있다. 하부노드인 v7부터 시작한다면 2단계(2~3단계) 2개의 노드(v7, v3)에서만 이벤트만 발생하고 v6부터 시작된다면 3단계(1~3단계) 3개 노드(v10, v6, v3)에서 이벤트가 발생한다. 이럴 경우, 위협수준은 공격경로 선택에 따라 공격 단계와 노드의 수에 의해 달라질 수 있다. 공격 트리는 공격 이

벤트가 발생하여 공격목적까지 달성하는 데 각 프로세스 단계별 위협수준을 확인하여 최대의 위협을 가할 수 있는 공격 경로를 선택해야 한다.



〔그림 4〕 공격 트리의 단계

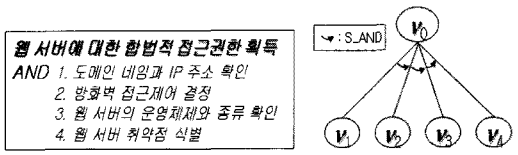
III. 해결방안

능동적인 사이버 공격 트리(Dynamic Cyber Attack Tree; D-CAT)는 위에서 제시한 기존의 공격 트리에 대한 문제점을 공격 트리의 애트리뷰트 관점에서 몇 가지 구성요소를 추가하여 해결하였다. 문제점들은 공격 트리의 애트리뷰트 중에 조합식을 재구성함으로써 좀 더 상세하게 공격 시나리오를 표현할 수 있고 공격 이벤트의 발생 순서를 인지할 수 있다. 또한, 위협발생수준을 추가하여 공격 이벤트 진행에 따라 공격의 위협수준을 알 수 있다.

첫 번째로 AND로 연결된 노드들의 공격 이벤트의 발생순서의 모호성은 조합식에 'Sequent_AND-composition'(이하 S-AND)을 추가함으로써 처음 공격 이벤트 발생 노드부터 마지막 노드까지 순차적으로 이벤트 발생 순서를 제시한다〔11,12〕.

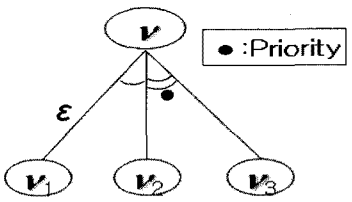
정의 1. composition = 〔S_AND, AND, OR〕; 순서를 나타내는 조합을 포함시킨다. node ν가 internal_nodes에 포함되고 노드들을 연결하는 앳지가 S_AND operation으로 연결되어 있으면 S_AND로 정의한다. 〔그림 2〕의 공격트리에서 노드 ν_0가 S_AND로 연결되어 있으면 첫 번째 노드 ν_1부터 마지막 노드 ν_4까지 순서대로 공격 이벤트가 발생한다는 것을 의미한다. 즉, $\nu_0=\{\nu_1,\nu_2,\nu_3,\nu_4\}$일 경우, $\epsilon=\langle\nu_0, S_AND\rangle$은 $\forall[(((\nu_1\wedge\nu_2)\wedge\nu_3)\wedge\nu_4)]\rightarrow\nu_0$로 표현할 수 있다. 〔그림 2〕의 공격 트리를 S_AND 조합을 사용하여 재구성하면 다음 〔그림 5〕와 같다.

두 번째로 조합식(AND, OR)으로 연결된 노드들의 공격 이벤트 발생 우선순위에 대한 불명확성은 조합식에 Label을 사용하여 우선순위를 나타낸다.



〔그림 5〕 S_AND 구성된 공격 시나리오와 트리

정의 2. 공격 트리에서 공격 우선순위를 표시하는 Label은 1비트의 "+"로 $\varepsilon=\langle\nu, composition(+)\rangle$와 같은 형식으로 표현한다. 〔그림 1〕에서 OR 조합으로 연결된 노드가 우선적으로 이벤트를 발생해야 한다면 $\varepsilon=\langle\nu_0,(AND, OR(+))\rangle$으로 표현하면 된다. Label "+"는 조합식으로 연결되어 있는 internal_ nodes 중에서 가장 먼저 공격 이벤트가 발생하는 조합을 표시한다. 〔그림 6〕의 공격 트리를 부울린 표현식으로 우선순위를 나타낸다면 $\exists[\nu_1\wedge\nu_2\dot{\vee}\nu_3]\rightarrow\nu_0$와 같으며, υ2와 υ3 노드 중 하나를 반드시 먼저 공격 이벤트를 발생시켜야 한다.



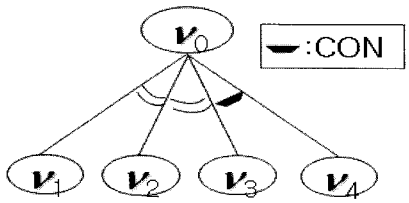
〔그림 6〕 Priority를 포함한 공격 트리

세 번째로 복잡하고 변형된 공격 시나리오의 표현을 위해 새로운 조합인 CONDITION-Composition(이하 CON) 조합〔13〕을 추가하면 된다.

정의 3. composition = 〔S_AND, AND, OR, CON〕으로 구성한다. node ν가 internal_nodes에 포함되고 internal_nodes 중에 CON operation에 의해 연결되어 있으면 CONDITION-composition으로 정의한다. 즉, CON 노드 ν는 AND와 OR 조합으로 공격방법 절차에 대한 표현이 불가능하거나 변형된 공격 이벤트가 발생할 경우를 대비하여 예비 노드로 활용한다.

〔그림 3〕에서 $\nu_0=\{\nu_1,\nu_2,\nu_3\}$일 때 새로운 공격이나 변형된 공격 이벤트가 발생하여 공격 트리를 확장할 필요가 있을 경우, $\epsilon=\langle\nu_0,(OR, CON)\rangle$은 $\exists[((\nu_1\vee\nu_2\vee\nu_3)\veebar\nu_4)]\rightarrow\nu_0$로 표현할 수 있다. 즉, υ1, υ2와 υ3 중 이벤트가 발생하여 기존의 공격 이벤트와 다른 양상으로 진행된다면 υ4를 CON 조합으로 새로운 공격

이벤트를 구성한다. CON 조합이 다른 조합과 다른 점은 S_AND, AND 또는 OR 조합은 인접한 노드에 적용되지만, CON 조합은 동일한 단계의 자식 노드들 중에 하나를 선택하여 적용한다.



〔그림 7〕 CON 조합이 추가된 공격트리

마지막으로, 공격단계별 위험수준은 위협발생확률(Threat Occurrence Probability, 이하 TOP)〔13,14〕을 애트리뷰트에 추가하여 나타낼 수 있다. 위협발생확률(TOP)은 어떤 부모노드의 공격목적을 달성하기 위해서 부모노드와 관련된 모든 공격 노드 대비 자식노드의 공격발생 비율을 의미한다. 일반적으로 TOP은 어떤 공격 트리의 총 노드가 ATN={N1, N2, N3, ..., Nn}이면서 부모노드 N1의 자식노드가 ATN1={n1, n2, n3}일 경우 N1의 TOP은 N1의 노드 수인 3을 총 노드수 n으로 나눈 값이 되는 것이다. 즉, 3/n이 된다. 그러나 OR 조합으로 구성된 노드들 경우에 모두 공격 이벤트가 발생하는 것이 아니라 그 중에 하나만 발생하기 때문에 모두 발생한다고 가정하면 안 된다. 또한, CON 조합은 적용될 경우에만 공격 이벤트가 발생하기 때문에 CON 조합으로 연결된 노드들은 이벤트가 활성화될 때에만 계산식에 포함한다. TOP은 다음과 같이 정의한다.

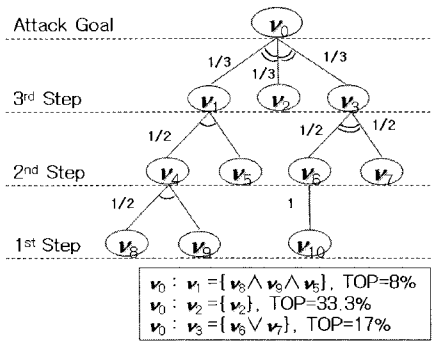
정의 4. Attack Tree = (ν, ε, θ, TOP)로 표현한다. TOP 계산은 3가지로 구분하여 계산한다.

(i) 자식노드가 오직 하나의 단일 노드(Leaf node)만 가지고 있다면 TOP=1

(ii) 자식노드가 S_AND와 AND 조합으로 구성되어 있다면 TOP=AND 조합 수 ÷ 총 노드 수

(iii) 자식노드가 OR 조합으로 구성되어 있다면 TOP=1/총 노드 수

〔그림 4〕에서 υ6은 단일 노드 υ0만 갖고 있기 때문에 υ6의 위협발생확률은 1이 된다. υ1은 υ4와 υ5가 AND 조합으로 구성되어 있어서 TOP은 AND 조합 수 1개, 총 노드 수 2개로 50%(1/2))의 TOP 값을 갖는다. υ3은 υ6와 υ7이 OR 조합으로 구성되어 TOP 값은 1/2로 50%가 되며 υ0까지는 17%(1/2×1/3)된다. 〔그림 7〕과 같이 루트 노드까지 계산한다면 υ0의 목적을 달성하기 위하여 υ1을 선택할 경우에 TOP 값은 8%(1/2×1/2×1/3)이다. 또한 υ1의 목적을 달성하기 위해서는 TOP 값이 25%이다.

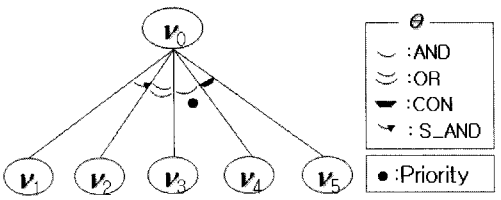


〔그림 8〕 공격 단계별 위협발생확률(TOP) 값

IV. 능동적인 사이버 공격 트리

4.1 능동적인 사이버 공격 트리 설계

기존의 공격 트리가 갖고 있던 제한사항을 해결한 능동적인 사이버 공격 트리(D-CAT)는 정점(ν), 앳지(ε), 조합(θ), 위협발생확률(TOP)로 이루어진다. 정점(ν)는 기존의 공격 트리와 동일하고 앳지(ε)는 자식 노드에서 공격을 수행할 때 우선순위를 나타내는 Label을 추가하였으며, 조합(θ)은 'OR, AND' 이외에 S_AND, CON을 추가하였다. D-CAT의 표현식은 D-CAT = $\langle \nu, \varepsilon, \theta, \text{TOP} \rangle$와 같으며, 각 구성요소에 대한 정의는 아래와 같다. 이 중에 기존의 공격 트리와 동일한 요소인 정점(ν)에 대한 정의는 생략한다. D-CAT = $\langle \nu, \varepsilon, \theta, \text{TOP} \rangle$를 공격 트리로 표현하면 다음 〔그림 9〕와 같다.



〔그림 9〕 D-CAT

① $\varepsilon \subseteq V \times V$ 는 공격 트리에서 앳지의 집합을 의미한다. ε는 $\langle \nu$, composition(+)$\rangle$ 형식인 튜

〔표 1〕 분산서비스거부공격 시나리오

- 목적: 서버의 동작 마비
- 공격대상: 서버
- 공격방법
  OR 조합: 1. UDP Flooding, 2. SYN Flooding, 3. ICMP Flooding, 4. HTTP GET Flooding(+)
- 사후조건: 공격대상 서버의 서비스 제공 불가

4. GET Flooding 공격
  - 목적: 웹서버의 동작 마비
  - 공격대상: PHP, ASP 등의 사이트 내 다이내믹 콘텐츠
  - 공격방법
    S_AND 조합: 5. TCP 연결, 6. 사이트내 다이내믹 컨텐츠 접속, 7. URL에 반복적 접근
  - 사후조건: 웹서버의 서비스 제공 불가

※ GET Flooding 공격 중 CC 공격으로 변형 가능
  S_AND 조합: 8. 같은 페이지 내 이미지 파일 등 대용량 파일 지속 다운로드 요청

플들의 집합을 의미한다. Label "+"는 조합으로 연결된 노드의 공격 이벤트 발생 우선순위를 나타낸다.

(i) $\nu \in$ internal_nodes and: ν는 internal_node(leaf_node 제외)에 포함된다.

(ii) Label(+) is a set of the priority of attack events: "+"은 공격 이벤트 우선순위 표시

(iii) composition = {S_AND, AND, OR, CON};

② 조합(θ)은 정점(ν)에서 internal_node에 포함되어 있는 노드들을 연결시키는 모든 앳지의 작동 모드를 정의한다.

(i) node $\nu \in$ internal_nodes이면서 자식노드들의 모든 앳지가 S_AND operation에 의해 연결되면 노드들의 공격 이벤트가 순차적으로 발생한다는 것을 의미하며 S_AND-composition으로 정의한다.

(ii) node $\nu \in$ internal_nodes이면서 자식노드들의 앳지 중에 CON operation에 의해 연결된 노드가 있다면 AND나 OR operation과 결합하여 공격 이벤트를 발생한다는 것을 의미하며, CON-composition으로 정의한다. CON으로 연결된 자식노드 ν는 다른 조합으로 연결된 노드들과 결합하여 표현이 불가능하거나 변형된 공격 이벤트가 발생할 경우에 예비노드와 결합시키는 역할을 한다.

③ TOP은 node $\nu \in$ internal_nodes에서 ν의 공격목적을 달성하기 위해서 ν 노드와 관련된 모든 자식노드 대비 조합식의 관계를 정의한다. 조합식에 따른 TOP 계산식은 아래와 같다.

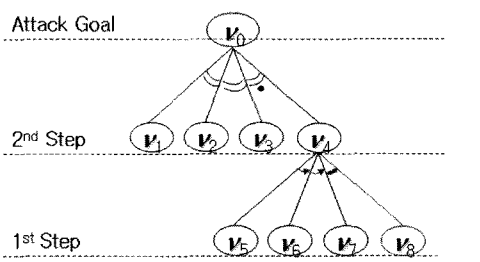
(i) 자식노드가 오직 하나의 단일 노드(Leaf node)만 가지고 있다면 TOP=1

(ii) 자식노드가 S_AND와 AND 조합으로 구성되어 있다면 TOP=AND 조합 수 ÷ 총 노드 수

(iii) 자식노드가 OR 조합으로 구성되어 있다면 TOP=1/총 노드 수

4.2 평 가

본 장에서는 분산서비스거부공격(DDoS)에 대한 공격 트리를 구성하여 D-CAT를 평가한다. DDoS 중에 HTTP GET Flooding 공격[15]은 트래픽을 거의 유발하지 않고 정상적으로 웹 접속을 하지만 사이트 내 다이나믹 콘텐츠에 대해 집중적으로 접속을 시도하여 웹 서버를 마비시키는 공격 방법이다. 또한, HTTP GET Flooding 변형공격으로 CC(Cache-Control) 공격[16]은 HTTP User-agent 헤더에 Cache-Control 값을 비정상적으로 조작하여 직접 공격대상의 URL을 호출하는 방법을 사용하여 해당



〔그림 10〕 분산서비스거부공격 트리

서버의 동작을 마비시킨다. 공격 시나리오는 [표 1]과 같다. 분산서비스거부공격 시나리오를 참조해서 공격 트리를 아래 [그림 10]과 같이 표현할 수 있다.

분산서비스거부공격은 무슨 프로토콜을 이용한 Flooding 공격방법을 결정해야 한다. [그림 10]에서는 UDP Flooding 공격 노드인 v1에서 HTTP GET Flooding 공격노드인 v4까지 있으나, 우선순위 Label이 v4에 있기 때문에 HTTP GET Flooding 공격을 선택한다. v4의 공격목적을 달성하기 위해서 D-CAT는 S-AND 조합식을 이용하여 v5에서 v7까지 공격 이벤트가 순차적으로 발생하도록 한다. HTTP GET Flooding 공격 중에 변형될 수 있는 공격은 CC 공격이다. CC 공격은 v5에서 v6까지 HTTP GET Flooding 공격과 동일한 공격 이벤트를 발생하나, 마지막에는 [표 1]처럼 다르게 발생한다. 기존의 공격트리에서는 공격 이벤트가 v6에서 v7로 전이가 되지 않으면 침입으로 인정하지 않을 수 있다. 그러나 D-CAT에서는 CON 조합식으로 '동일한 웹 페이지내 대용량 파일 다운로드 요청' 이라는 공격 이벤트(v8)를 예비노드로 설정하여 HTTP GET Flooding 공격 절차에서 벗어날 경우에 결합하여 새로운 변형된 공격을 탐지할 수 있도록 하였다. HTTP GET Flooding 공격은 위협발생확률이 16.7%로 UDP, SYN, ICMP Flooding 공격의 위협발생확률을 산출하여 비교함으로써 위협수준을 확인할 수 있다.

V. 결 론

본 논문에서는 기존의 공격 트리가 갖고 있던 제한사항을 식별하고 그에 따른 해결책을 제시함으로써 개선된 능동적인 공격 트리를 제안하였다. 기존의 AND 조합으로 연결된 노드들의 공격 이벤트는 어떤 노드부터 순차적으로 발생해야 할지, 여러 개의 조합으로 연결되어 있는 노드들 중에서 어떤 노드부터 이벤트를 발생시켜야 할지 알 수가 없다. 또한, 복잡하고 변형된 공격은 AND와 OR 조합식으로 표현하는 데는 한계가 있다. 마지막으로 자식노드부터 부모노드, 최종노드까지 공격 이벤트가 진행되면서 공격의 위험수준을 알 수가 없다.

능동적인 사이버 공격 트리(D-CAT)는 노드들의 공격 이벤트 순서와 우선순위는 조합식에 S_AND 조합과 우선순위를 나타내는 조합 표시로 해결하였다. 제한된 공격 시나리오를 표현하기 위해서는 조합(θ)에 CON 조합식을 추가하여 표현의 유연성을 증가시켰다. 공격 위험수준은 위협발생확률(TOP)을 공격트리 표현식에 포함하여 공격 진행 단계별로 위험의 정도를 알 수 있게 하였다.

참고문헌

[1] Aleksandr Matrosovl, "Stuxnet Under the Microscope," www.eset.com/ resources /white-papers /Stux-net_ under_the_Microscope.pdf, 2010.

[2] 엄정호, 정태명, 한영주, 박선호, "사이버 공격과 보안기술," 홍릉과학출판사, 2009년.

[3] Vineet Saini, Qiang Duan and Vamsi Paruchuri, "Threat modeling using attack trees," Journal of Computing Sciences in Colleges, Vol. 23, Issue 4, pp. 124-131, Apr. 2008.

[4] S. Mauw and M. Oostdijk, "Foundations of attack trees," In Eighth Annual International Conference on Information Security and Cryptology, LNCS 3935, 2006.

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

[6] Ariel Futoransky, Luciano Notarfrancesco, Gerardo Richarte and Carlos Sarrautel, "Building computer network attacks," Technical report, Core Labs, Core Security Technology, 2003.

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

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

[9] 이종엽, 윤미선, 이훈, "DoS 공격의 유형 분석 및 탐지 방법," KNOM Reveiw, Vol. 6, No. 2, pp. 33-40, Feb. 2004.

[10] 엄정호, 박선호, 정태명, "네트워크 취약성 분석을 위한 확장된 사이버 공격 트리에 관한 연구," 디지털산업정보학회논문지, 6(3), pp. 49-58, 2010년 9월.

[11] 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, pp. 234-243, September 2007.
[12] Parvaiz Ahmed Khand, "System level Security modeling using Attack trees," Proceedings of the 2nd International Conference on Computer, Control and Communication, pp. 1-6, 2009.
[13] Jung ho Eom, Seon ho Park, Young ju Han and Tai M. Chung, "Active Cyber Attack model for Network system's Vulnerability Assessment," International Conference on Information Science and Security, pp. 153-158, Jan. 2008.
[14] Indrajit Ray and Nayot Poolsapassit, "Using Attack Trees to Identify Malicious Attacks from Authorized Insiders," 10th European Symposium on Research in Computer Security, LNCS 3679, pp. 231-246, 2005.
[15] "국내 주요사이트 대상 분산서비스거부공격 분석 보고서," 한국정보보호진흥원, 2009년 7월.
[16] "DDoS 공격 비상, 어떻게 대처할 것인가? -7.7 DDoS 공격 유형 분석 및 대응방안-," 시스코 시스템즈 코리아, 2009년 7월.

〈著者紹介〉

엄 정 호 (Jung-ho Eom) 정회원
1994년 3월: 공군사관학교 항공공학과 졸업
2003년 2월: 성균관대학교 컴퓨터공학과 석사
2008년 2월: 성균관대학교 컴퓨터공학과 박사
2011년 2월: 성균관대학교 정보통신공학부 BK21 연구교수
2011년 3월~현재: 대전대학교 군사학과 교수
〈관심분야〉 사이버전, 사이버공격, 접근제어, 네트워크 보안