情報保護學會論文誌
第20卷 第1號, 2010. 2

Role Based Petri Net : 공격 시나리오의 효율적 설계를 위한 역할 기반 표현 모델

박 준 식,† 조 재 익, 문 종 섭‡

고려대학교 정보경영공학전문대학원

Role Based Petri-Net : Role Based Expression Model for an Efficient Design of Attack Scenarios

Jun-Sik Park,† Jae-ik Cho, Jong-sub Moon‡
Graduate School of Information Management and Security, Korea University

요 약

공격 시나리오의 그래프 표현은 서버의 취약성 분석 및 공격의 방어를 위한 설계에 필수적인 방법이다. 이를 위해 다양한 요구사항 분석 모델이 이용되고 있으나, 복잡한 시나리오간의 결합을 표현할 수 있는 모델은 제한적이다. 본 논문에서 제안하는 역할 기반 페트리 넷(Role Based Petri Net)은 동시성과 시각적인 장점을 가진 페트리 넷을 역할 기반으로 구성하여 효과적 표현 모델을 제공하고 알려지지 않은 공격에 대한 시나리오를 효율적으로 표현할 수 있다.

ABSTRACT

Graph expression of attack scenarios is a necessary method for analysis of vulnerability in server as well as the design for defence against attack. Although various requirement analysis model are used for this expression, they are restrictive to express combination of complex scenarios. Role Based Petri Net suggested in this paper offer an efficient expression model based role on Petri Net which has the advantage of concurrency and visuality and can create unknown scenarios.

Keywords: Petri Net, Security, Design, Analysis, RBAC

I. 서 론

공격 시나리오를 표현하기 위해 활용된 분석 설계 기법으로 상태 다이어그램, 결함 트리, 페트리 넷 등이 활용되고 있다. 상태 다이어그램은 상태 진행을 단순하게 표현하기 위해 모델링 되었으나 동시성이 제공되지 못하여 행위 모델링인 공격 시나리오에 적절하지 못하다. 결합 트리는 어떤 특정한 사고에 대하여 그 사고의 원인을 분석하여 정량적으로 안정성을 평가 진단하기위한 목적으로 생성되어 1) 간단하고 2) 사용하기 쉬우며 3) 직관적으로 이해를 제공하지만, 노드 간에 복잡한 설명과 연결이 불가능하여 다양한 시나리오 확장이 불가능하다.[1]

페트리 넷은 추가 확장이 용이하며 동시성이 제공되는 설계 모델로써 공격 시나리오의 표현에 적절하다. 그러나 복잡한 시나리오에 대한 상태 전이는 제한적이다. 본 논문에서 제안하는 Role Based Petri Net(이하 RBPN) 모델은 알려지지 않은 공격에 대한 시나리오를 효율적으로 표현하고 다양한 공격 시나리오의 전이를 표현하기 위해 페트리 넷을 역할 기반으로 표현하고 그에 따른 상태 전이를 제안한다.

접수일(2009년 5월 18일), 게재확정일(2009년 12월 12일)
† 주저자, qweruio@daum.net
‡ 교신저자, jsmoon@korea.ac.kr

본 논문은 2장에서 페트리 넷의 배경과 문제점을, 3장에서는 문제점을 해결하기 위해 제안하는 모델을 설명한다. 4장에서는 제안 모델의 표현 사례를 통해 그 효율성과 역할 플레이스(Role Place : 이하 RP)를 통해 신규 시나리오의 생성 가능성을 확인한다. 마지막으로 5장은 본 논문의 결론으로 구성된다.

II. 관련연구

2.1. 페트리 넷의 일반적 개념

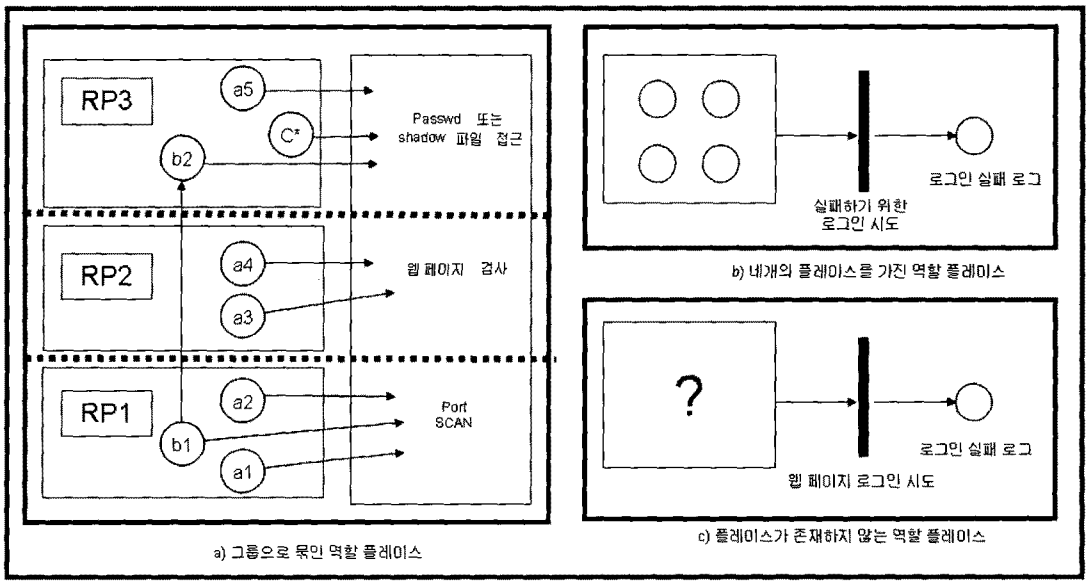
1960 년대 C.A Petri에 의해 처음 개발된 페트리 넷 모델[2]은 소규모 사회 활동이나 유동 관계 등 여러 가지 행위를 분석하고 표현하기 위해 개발된 방법이다. 이는 결함 트리와 함께 행위 모델로써 설계와 구현을 위한 모델로 다양하게 연구되고 있다.[3] 페트리 넷 모델의 표현 방식은 시나리오의 위치에 따른 상태와 다른 상태로 이동하기 위한 행위를 그래프 표현을 통해 기술하는 방식으로, 플레이스, 트랜지션, 아크, 그리고 토큰 네 가지로 구성되어 1) 시각성 2) 의존 속성 3) 동시성 제공 등의 특징을 제공한다.[4]

2.2 페트리 넷의 종류

Colored 페트리 넷(이하 CPN)은 가장 널리 알려진 모델로서 토큰에 따라 각기 다른 값을 부여할 수 있다.[5] 객체 지향 페트리 넷[6]은 방대해진 요구사항에 대해 효율적인 명세를 위해 모듈화 및 객체화 개념을 도입하여 정보 은닉, 추상화, 상속성, 메시지 패싱, 다형성 등을 제공하였다. 그 밖에 Finite Capacity 페트리 넷[7]은 토큰의 개수를 한정하였고, Inhibitor Arc 페트리 넷[8]은 플레이스에 토큰이 존재할 경우 또 다른 트랜지션의 진입을 금지하도록 하는 등 여러 모델들이 제안되고 있다. 또한, Role based extended petri net 모델[9]에서 항공사의 관제탑과 파일럿의 행위를 각각 역할별로 작성하여 효율적으로 활용할 수 있음을 확인하였다. 본 모델에서 제안하는 RBPN 은 페트리 넷을 기반으로 플레이스들을 역할에 따른 플레이스로 재 표현하며 CPN 및 객체 지향 페트리 넷 등 다양한 모델의 기능을 유지할 수 있다.

2.3. 페트리 넷의 문제점

지금까지 연구된 페트리 넷은 대부분 공격 시나리오를 표현하기 위해 다음과 같은 어려움이 존재한다. 1) 플레이스의 재사용이 많은 경우 복잡성이 가중된다, 2) 플레이스의 추가 삭제가 빈번한 시나리오의 경우 그래프의 재구성으로 관리가 어렵다. 3) 플레이스에 대한 트랜지션이 많은 경우 공격 시나리오의 이해도가 떨어진다. 4) 알려지지 않은 공격 상태에 대한 플레이스 표현의 어려움으로 신규 공격 시나리오의 예



[그림 1] 역할 플레이스 모델

측성이 결여된다. 본 논문에서 제안하는 역할 기반 플레이스의 표현은 이러한 문제점을 보완할 수 있다.

III. 제안하는 방법

RBPN의 생성 목적은 공격 시나리오를 효율적으로 설계하기 위해 페트리 넷의 플레이스를 역할 기반으로 확장 변형한 모델이다. RBPN의 핵심 표현인 RP는 역할 기반으로 재 표현한 플레이스를 의미한다. 이는 논리적으로 공격의 상태를 권한에 따라 또는 공격 결과의 형태에 따라 역할기반으로 재구성하였으며, 물리적으로 타 시나리오로 이동이 가능한 플레이스들의 조합을 RP 형태인 사각 블록 모양으로 표현하여 기존 플레이스와 구분된다. 이는 권한 상승을 효율적으로 파악할 수 있으며, 공격의 변화에 따른 공격 시나리오 설계 변경이 용이하며, 알려지지 않은 공격 상태를 추상적으로 표현할 수 있는 특징을 가진다. 또한, 기존 페트리 넷 표현의 트랜지션을 표현을 동일하게 적용하며 다양한 공격 시나리오의 흐름을 효율적으로 표현 할 수 있다.

그림 1의 a 는 RP를 표현하는 개념도이다. 각기 다른 공격으로부터 동일한 공격 플레이스로 진행하는 플레이스가 위와 같이 다수 존재할 경우, 동일한 트랜지션을 가진 플레이스들에 대해 RP(n) 와 같이 그룹화 할 수 있다. 그림 b는 로그인의 실패로 로그를 발생시고자하는 경우, 가능한 플레이스들을 RP로 그룹화한 예이다. 그룹 내부의 플레이스는 각각 1) 사용자 데이터베이스의 권한 변경이 이루어진 경우 2) 데이터베이스가 변조된 경우 3) 데이터베이스가 삭제된 경우 4) 데이터베이스에 물리적 결함이 생성된 경우와 같이 구성될 수 있다. 또한, 그림 c와 같이 특정 플레이스가 발견되지 않았지만 시나리오 분석가에 의해 특정 역할을 예측 표현하여 다양한 공격의 융합을 통해 새로운 시나리오 경로가 제시될 수 있다.

3.1 공격 분석 절차와 활용

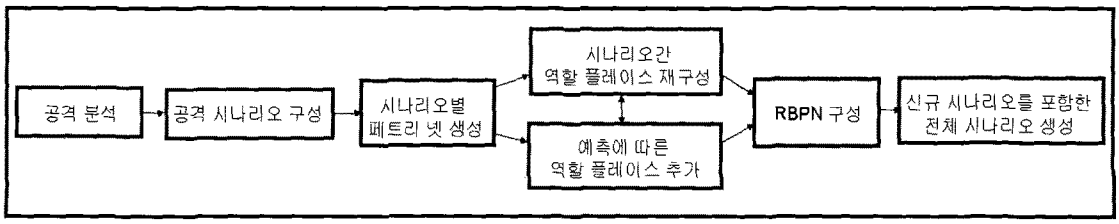
그림 2는 RBPN을 구성하고 이를 통해 전체 시나리오를 생성하는 순서도이다. 단일 공격 시나리오를 분석하여 페트리 넷을 구성한 이후 여러 시나리오를 재구성 또는 예측에 따른 추상 표현을 통해 RBPN을 생성할 수 있다. 이는 전체 시나리오를 파악하는 기반이 되어 서버의 방어 시스템 구축, 취약성 분석, 침입 탐지 시스템, 침입 방지 시스템 등의 설계 모델로 공격 분석가 또는 설계자에게 활용 될 수 있다.

3.2 RP의 시맨틱에 따른 구성요소

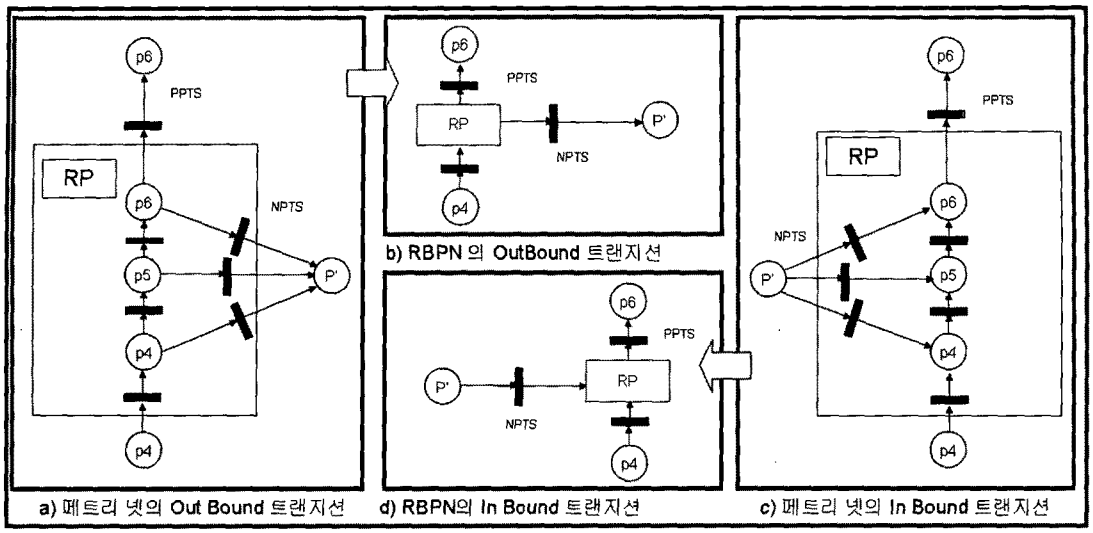
정의 1) RP는 하나 이상의 플레이스를 가진 역할 플레이스로서, 내부의 플레이스가 하나이면 일반 플레이스와 동일하다. RP는 그림 3과 같이 식별자, 부모 정보, 이웃 시나리오와의 전이 정보 그리고 플레이스 타입으로 구성 되며 아래와 같이 정의한다.

정의 2). IP (Identification Place) 는 RP 에 대한 유일한 식별자이며, IP={ipn, ipid, ipts} 로 구성된다. ipn (Identification Place Name) 은 RP 내부에 존재하는 플레이스 세트 이름이며, ipid (Identification Place Identifier) 는 RP 에 해당하는 식별자이다. ipts (Identification Place Transition Set) 는 RP 내부의 트랜지션과 플레이스를 순차 리스트 형태로 나열한 세트이며, 개수에 제한이 없다.

정의 3) PPTS(Parent Place Transition Set) 은 RP와 상위 플레이스와의 관계를 의미하고, 내부 플레이스 중 일부 또는 최상위 플레이스가 외부의 부모 플레이스로와의 관계를 정의한다. PPTS={sp, tp, t, a} 로 구성된다. sp(Source Place) 는 부모 플레이스로 전이 가능한 RP 내부의 플레이스 집합이다. tp(Target Place) 는 RP 외부의 플레이스 집합으로 전이 되는 RP 의 상위에 존재하는 부모 플레이스 집합이다. t(Transition) 는 RP 내부의 최



[그림 2] RBPN을 통한 신규 시나리오 생성 순서도



〔그림 3〕 입력 방향 역할 플레이스와 출력 방향 역할 플레이스

상위 플레이스 와 부모 플레이스와의 트랜지션이다. a (Arc) 는 해당 RP와 상대 플레이스와의 연결 정보이다.

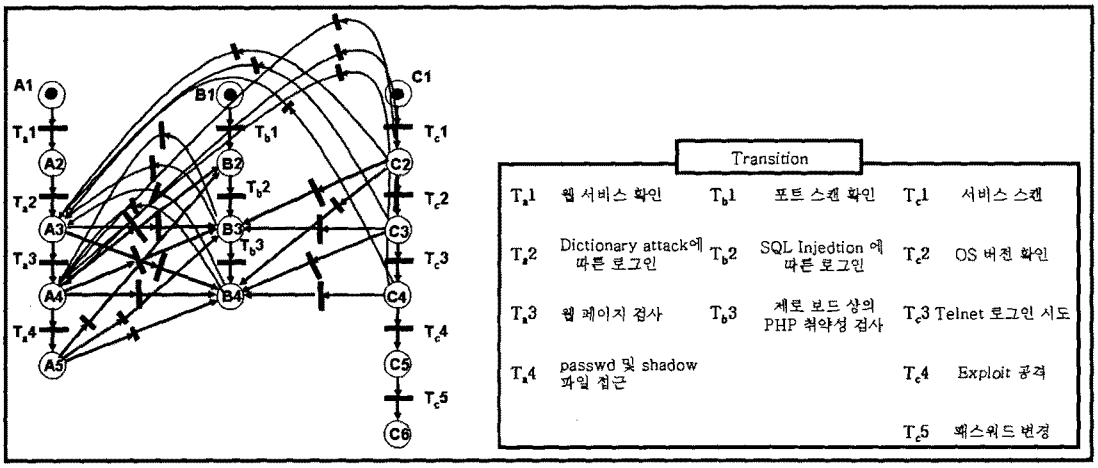
정의 4) NPTS 는 (Neighbor Place Transition Set) 는 RP 내부의 플레이스와 타 시나리오와의 관계를 의미하고, RP 와 결합 가능한 이웃 플레이스와의 관계를 정의한다. NPTS={sp, tp, t, a}와 같이 구성되어 리스트 형태로 보유하고 있다. 이는 RP 내부의 구성요소는 PPTS와 동일한 형태로서, 모든 플레이스들이 이웃 플레이스로 트랜지션이 가능하다.

IV. 기존 모델과의 비교 분석

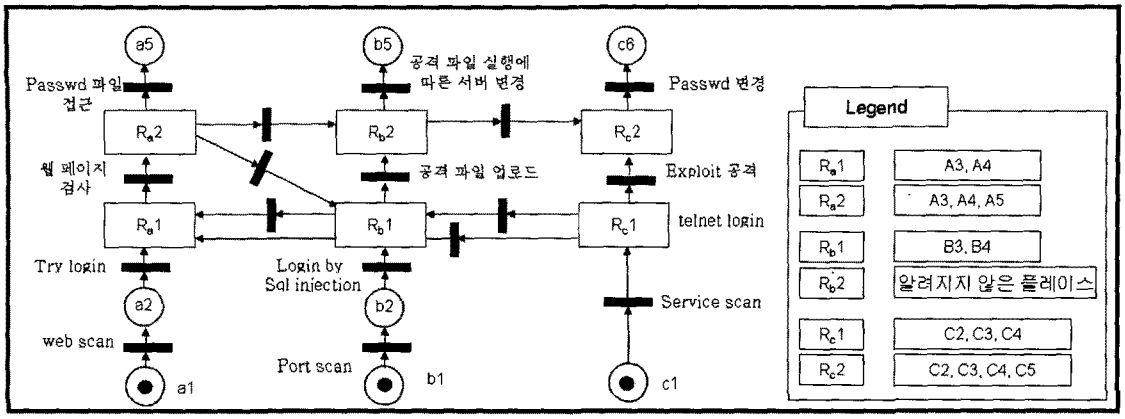
RBPN은 페트리 넷의 역할에 따른 표현으로 플레이스와 트랜지션의 개수를 줄여 복잡한 시나리오를 단순하게 표현할 수 있는 특징을 가지며 페트리 넷과 동일한 시나리오 경로를 제공할 수 있다. 이를 위해 경로 효율성 분석과, 동일한 경로 검색 시나리오를 확인하며, 추가 시나리오 생성 가능성을 확인한다.

4.1 경로 효율성 검색

그림 4는 제안된 모델과의 기존 모델과의 비교 분석을 위해 1) 패스워드 파일 접근 공격 2) 제로보드의 취약성 검사 3) Exploit 공격에 따른 패스워드 변조



〔그림 4〕 기존 페트리 넷 표현에 따른 시나리오의 유기적 결합



〔그림 5〕 RBPN 표현 방법에 따른 표현과 발견되지 않은 역할 플레이스

공격 시나리오를 페트리 넷으로 표현한 예이다. 이와 같이 페트리 넷은 시각적으로 복잡하여 그 기능이 상실된다. 그림 5는 그림 4를 RBPN 에 따른 표현으로 변환한 것이다. 이는 동일한 트랜지션을 가진 플레이스를 Ra1, Ra2, Rb1, Rc1, Rc2와 같이 재구성하여 시각적으로 트랜지션 개수가 축약되었다.

RBPN이 페트리넷에 비해 효율적임을 확인하는 경로 효율성 검색은 트랜지션 개수와 플레이스 개수를 통해 확인한다. 트랜지션은 크게 PPTS와 NPTS의 개수로 구분할 수 있다. PPTS의 트랜지션은 RP 내부의 플레이스 개수만큼 축소되며, NPTS의 트랜지션의 차이(Neighbor Transition Number Distance)는 (1)과 같이 변경 전과 후의 트랜지션 개수를 계산할 수 있다. RP의 플레이스의 조합이 NTPS의 개수보다 많기 때문에 페트리 넷의 트랜지션 개수에 비해 높은 효율성을 검증할 수 있다.

$$\begin{aligned} NTND &= \text{변경 전의 트랜지션 총합} \\ &\quad - \text{RP의 NPTS 개수} \qquad (1) \\ &= \sum_{i \in RP} (|sp_i| \times |tp_i| - |NTPS|) > 0 \end{aligned}$$

RP의 개수는 플레이스의 개수(n)의 조합에 의해 2n -1 개까지 생성이 가능하다. 하지만 RP는 일반적으로 공격권한을 획득하면 특정 플레이스의 하위 레벨로는 모두 전이될 수 있다. 이와 같이 RP는 특정 플레이스의 하위 플레이스를 포함하게 되므로 플레이스 개수(n)이상의 RP를 생성하지 않는다. 만일 타 시나리오와의 불가피한 권한 상승 조합이 생성되더라도, 타 시나리오 개수 m 에 따라 최대 (m -1) × n (=t) 개를 넘지 않는다. 이 경우 RP 내부에 플레이스는 2개 이상이므로 트랜지션 개수는 2t -t (=t) 개로 축소된다. 이처럼 플레이스 증가 수와 동일하게 트랜지션 개수가 축소된다 하더라도 트랜지션 표현은 플레이스에 비해 시각적이지 않으므로 제안된 표현이 더욱 효율적이다.

4.2 시나리오 경로 검색

표 1은 페트리 넷의 그래프 표현을 통해 출력 방향에 대한 시나리오 경로를 생성하는 주요 알고리즘이다. 시나리오 경로는 토큰에 의해 발화되어 트랜지션에 따라 경로가 추가되며, 타 시나리오로 트랜지션이 가능한 NPTS가 존재하면 저장된 경로가 복제되어 새로운 경로를 생성하게 된다. PushPath는 추가될 플레이스를 인자로 반복적인 호출을 통해 시나리오 경로를 생성하는 함수이다. 이때 입력 받은 플레이스가 RP 형태라 하더라도, 내부 플레이스만큼 경로를 복제하여 페트리 넷을 통한 시나리오 경로와 동일한 결과를 가질 수 있는 알고리즘이 된다. 본 알고리즘의 간략화를 위해 ClonePath를 정의하였다.

* ClonePath(path, place) : 현 플레이스의 이전 플레이스에 추가적으로 새로운 경로를 생성한다.

4.3 신규 생성 시나리오

그림 5에서 Rb2 는 발견되지 않은 공격을 예측한 RP로 특정 파일의 업로드가 가능한 상태를 추상적으로 생성하였다. 이 RP가 로그인 된 웹 페이지 상에서 가능하다면, Rb1 과 Ra2로부터 트랜지션이 가능하

[표 1] 공격 시나리오 생성 알고리즘

```
PushPath (Input - scenario : 시나리오 경로,
                place:추가될 플레이스)
scenario <- add to place
while (transition <- transition set of place) {
  if (transition type & PPTS) then
    PushPath (path, place of PPTS)
  else if ((transition type & NTPS)
      && (arc & outbound) ) then
    if (place.type & Place type) then
        path = ClonePath (path, place);
        PushPath (place of NPTS)
    else if (place_type & RP type) then
       while (place <- sp in RP)
        {path = ClonePath (path, place);
        PushPath (place of NPTS)}
```

다. 또한, 특정 권한을 획득된 것으로써 로그인에 따른 Exploit 공격 시나리오와 이어질 수 있을 것으로 예측되어 Rc2 시나리오로 트랜지션이 가능하다. 추가 공격으로 공격 파일 업로드를 시도하고 해당 파일을 실행할 수 있다면 웹페이지 변조 공격이 가능할 것으로 예측되어 알려진 공격 플레이스가 추가되었다. 추가된 RP는 모든 시나리오가 유기적으로 결합가능하고, RP 내부 플레이스의 개수에 무관하게 유연한 시나리오를 생성할 수 있다.

V. 결론

이 논문에서는 다각적인 공격에 대한 시나리오를 효율적으로 표현하기 위해 정형화한 모델로써 RBPN을 제안하였다. 이는 요구사항 모델로써 공격의 추가 확장의 용이성, 시각성, 동시성의 장점을 가진 행위 모델인 페트리 넷을 기반으로 하였으며, 보다 복잡한 공격을 표현하기 위해 RP를 제안하였다. RBPN의 효율성을 검증하기 위해 서버 공격의 시나리오를 표현해 보았으며, 신규 공격을 어떻게 표현 할 수 있는지 확인하였다. 향후에는 RBAC 의 개념을 확장 적용하여 계층 및 세션 형태, 전후 상황에 따른 동적인 구성 형태의 모델 등으로 시나리오 생성 뿐 아니라 요구사항 분석에 다양한 이점을 제공할 수 있을 것으로 예측된다. 그러나 역할 형태로 보다 효과적인 표현 모델을 제공한다 하더라도 무한히 다양하고 많은 시나리오와 그에 따른 트랜지션 결합에는 시각적인 표현의 한계가 존재하여 그에 따른 연구가 필요하다.

참 고 문 헌

[1] R. Beresh, J. Ciufo, and G. Anders, "Basic fault tree analysis for use in protection reliability," International Journal of Reliability and Safety, vol. 2, no. 1/2, pp. 64-78, Oct. 2008.

[2] C. Girault and R. Valk, "Petri Nets for Systems Engineering," Springer-Verlag, Secaucus. NJ. USA, 597, 2002.

[3] G. Helmer, J. Wong, M. Slagell, V. Honavar, L. Miller, Y. Yang, and R. Lutz, "Software Fault Tree and Colored Petri Net Based Specification, Design and Implementation of Agent-Based Intrusion Detection System," Int. J. Information and Computer Security, vol. 1, no. 1/2, pp. 109-142, Jan. 2007.

[4] T. Murata, "Petri Nets: Properties, Analysis and Applications," Proceedings of IEEE, vol. 77, no. 4, pp. 541-580, Apr. 1989.

[5] K. Jensen, "Coloured Petri nets: basic concepts, analysis methods, and practical use, Volume 3," Spinger-Verglag, Berlin, 265, 1997.

[6] 홍장의, 윤일철, 배두환, "객체지향 페트리 넷을 이용한 계층적인 요구사항의 명세 및 검증," 정보과학회논문지, 27(2), pp. 157-167, 2000년 2월.

[7] Y. Ru and W. Wu, "Finite Capacity Place Method Based Deadlock Prevention Algorithm," Journal of System Simulation, vol. 15, pp. 59-62, Aug. 2003.

[8] P. Baldan, N. Busi, A. Corradini, and G.M. Pinna, "Functorial concurrent semantics for Petri nets with read and inhibitor arcs," Proceedings of the 11th International Conference on Concurrency Theory, LNCS 1877, pp. 442-457, 2000.

[9] F.D.J. Bowden and M. Davies, "Application of a Role-Based Methodology to Represent Command and Control Processes Using Extended Petri Nets," IEEE International Conference on Systems, Man and Cybernetics, Orlando, Florida, USA, pp. 4348-4353, Oct. 1997.