논문 2011-48CI-3-9

CPN 기반의 침입방지시스템 보안모델의 안정성 검증

(Secured Verification of Intrusion Prevention System Security Model Based on CPNs)

이 문 구*

(Moon-Goo Lee)

요 약

침입방지시스템은 내부 시스템 보안 또는 외부 공격의 문제를 해결하기 위한 중요한 솔루션이다. 이러한 침입방지시스템을 도입 시 가장 우선적으로 고려해야 될 사항으로는 다양한 기능보다 안정성이다. 본 논문은 침입방지시스템 보안모델의 사용자 인증기능에 대한 안정성 검증을 위하여 칼라 페트리 네트를 이용하였다. CPN은 분산되어있고, 동시 발생적이며, 결정적 또는 동기화 방식의 비결정적인 시스템들에 대하여 그래픽적인 모델링 언어로 표현이 가능하다. 이런 칼라 페트리 네트는 각 처리 단계에 대하여 모든 가능한 상태와 발생 그래프로 표현된다. 침입방지시스템 보안 모델의 안정성은 칼라 페트리 네트를 이용한 모든 상태표현과 발생그래프의 분석결과가 무한반복 혹은 교착상태가 없으므로 검증되었다.

Abstract

Intrusion prevention systems (IPS) are important solution about solved problems for inside system security or outsider attacks. When introduce this system, first consideration item is secured rather than multiple function. Colored Petri Nets (CPNs) used that in order to secured verification for user authentication function of intrusion prevention system security model. CPNs is a graphical modeling language suitable for modeling distributed, concurrent, deterministic or non-deterministic systems with synchronous. Like these CPNs was expressed every possible state and occurrence graph. Secured of IPS security model was verified because expression every state using CPN tool and as a result of analyzing the occurrence graph was without a loop or interruption.

Keywords : 컬러 페트리 넷(Colored Petri Nets), 침입 방지 시스템(Intrusion prevention systems), 사용자 인증기능(User Authentication), 발생 그래프(Occurrence graph)

Ⅰ. 서 론

기업의 관문에 설치된 보안의 3대 요소인 AAA(Authentication : 인증, Authorization : 권한 부여, Accounting : 계정관리)를 네트워크에 적용한 침입방지 시스템(Intrusion Prevention System)은 필수적인 보안 솔루션 중의 하나이며, 이러한 침입방지시스템을 도입 시 가장 우선적으로 고려해야 될 사항으로는 다양한 기능보다 안정성과 편이성이다[5~6]. 침입방지시스템에서 요구되는 보안기능은 외부침입에 대한 지속적인 탐지와 침입이라고 판단되는 상황에 대한 침입차단 기능을 수행하기 위하여 반드시 요구되는 보안 기능으로는 강력한 로깅(logging)과 사용자 인증기능이다. 그러므로 본 논문에서는 침입방지시스템의 인증기능에 대한 안정성을 검증하였다. 안정성 검증을 위해서는 칼라 페트리 네트(Colored Petri Nets : CPNs)의 발생 그래프(Occurrence graph)와 상태불변식(Place Invariant)중에서 발생 그래프(Occurrence graph)를 이용하였다[1, 9, 11].

본 논문의 구성은 다음과 같다. Ⅰ장 서론에서는 연구

* 평생회원, 김포대학 IT학부 인터넷정보과
(Div. of IT, Dept. of Internet Information, Kimpo College)
※ 이 논문은 2011학년도 김포대학의 연구비 지원에 의하여 연구되었음.
접수일자: 2011년2월21일, 수정완료일: 2011년4월30일.

목적을 기술하고, Ⅱ장 본론에서는 제안하는 침입방지시스템의 보안모델과 CPN 기반의 안정성검증을 기술하였으며, Ⅲ장은 결론 및 향후 연구방향을 제시하였다

Ⅱ. 본 론

1. 제안하는 침입방지시스템의 보안기능

가. 제안하는 침입방지시스템 보안모델

기존의 침입차단시스템은 차단기능이 OSI 7모델에의 3계층인 네트워크계층(Network Layer)과 4번째 계층인 전송계층(Transport)에서 처리되기 때문에 처리속도가 빠르고, 사용자에게는 전송의 투명성을 제공한다는 장점을 갖고 있다. 그러나 TCP/IP 패킷의 헤더는 쉽게 조작이 가능하여 외부 침입자가 이러한 패킷의 정보를 조작한다면 내부시스템과 외부시스템이 직접연결 된다. 또한 ftp, mail 등에 의해서 유입되는 바이러스가 감염된 파일이 전송될 경우 잠재적으로 위험한 데이터에 대한 분석이 불가능하다[0]. 그리고 침입차단시스템이 컴퓨터 시스템 및 네트워크에 가해지는 내부 혹은 외부의 침입 행위를 실시간으로 탐지하고 방어하는 기능은 극히 제한되어 있다.

이에 제안하는 침입방지시스템은 기존에 사용되고 있는 응용서비스와 새로운 서비스에 대한 연동을 쉽게 할 수 있도록 유연성 있는 설계를 하였으며, 침입방지시스템의 보안모델은 실시간으로 내부 시스템 및 외부 네트워크로의 침입 여부를 탐지하고 탐지 엔진의 규칙(rule)을 기반으로 이상 상황에 대한 경고 메시지 전송과 제어 그리고 자동 차단 기능을 수행 할 수 있도록 설계하였다. 이러한 보안기능이 실행되기 위해서 시스템은 실시간 감시(monitoring)가 이루어지고, 사용자가 서버에 접속한 후에도 지속적으로 침입탐지엔진의 구동에 의해 이상 상황의 발생을 탐지하게 되고, 만약 침입이나 이상 상황이 발생한 경우는 차단엔진의 동작에 의해 해당 IP를 차단하거나 경고 및 제반 조치를 취할 수 있도록 하였다. 이러한 처리들을 위해서는 시스템의 실시간 로깅(logging)과 강력한 사용자 인증기능이 제공되어야 한다. 그러므로 본 논문에서는 설계한 침입방지시스템 모델의 강력한 인증기능에 대한 안정성 검증을 실시하였다. [그림 1]은 제안하는 침입방지시스템의 보안모델을 도식화 한 것이다.

나. 침입방지시스템의 사용자 인증기능

침입방지시스템의 강력한 인증기능을 위해서 국가공개키 기반구조(NPKI : National Public Key Infrastructure)의 공인인증과 일회용 패스워드(One-time password)를 병행 한다. 시스템에서 사용되는 일회용 패스워드 기능은 오직 한 번의 사용자 인증에만 유효하며, 동일한 패스워드의 재사용은 불가능하기 때문에 불완전한 네트워크상에서의 가능한 도청 및 재시도 공격에 안전하다[11].

인증기능 처리과정은 [그림 2]와 같이 인증(auth) 처리 모듈을 시작하기 위한 처리과정, 인증 데이터베이스를 open하는 처리 모듈, 데이터베이스에서 자료를 선택하는 처리 모듈, 인증 데이터베이스를 close하는 과정의 처리 모듈, 인증된 결과에 대한 메시지를 사용자에게

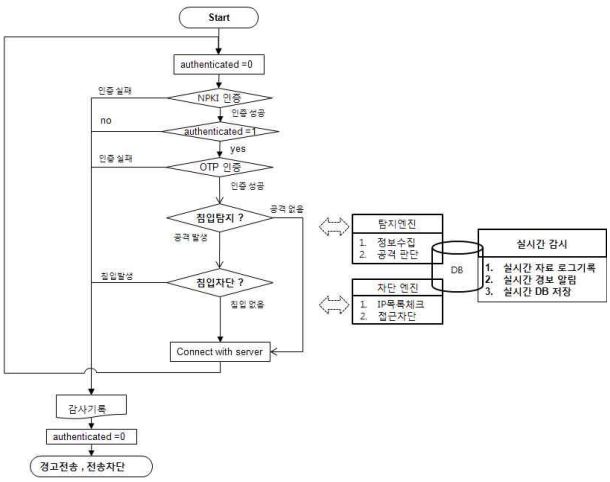

그림 1. 침입방지시스템의 보안모델
Fig. 1. Security Model of IPS.

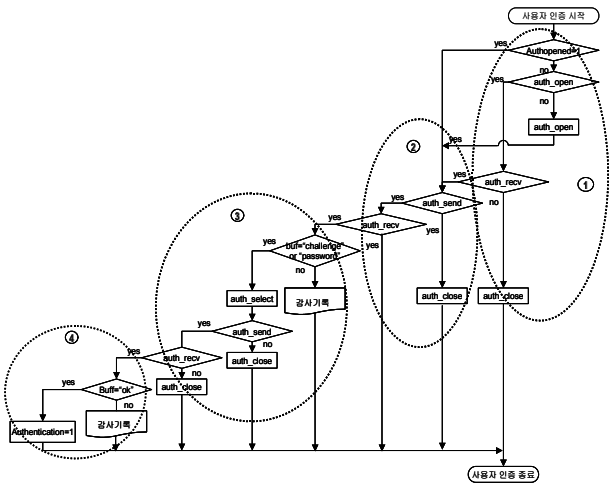

그림 2. 사용자 인증 기능 처리과정
Fig. 2. Process of User Authentication.

전송 처리하는 모듈, 인증 결과를 응답하는 처리 모듈로 구성된다.

2. CPN기반의 안정성검증

가. CPN의 도입목적과 발생그래프

침입방지시스템 보안모델의 인증기능에 대한 안정성을 검증하기 위하여 CPN(Coloured Petri Net)을 도입하게 된 주요 목적은 다음과 같다[1][2][3][9][11].

- 시스템의 각 기능에 대한 흐름이 그래픽으로 표현이 가능하다.
- 모델화 한 시스템의 각 진행단계를 논리형식에 맞는 구문의 표현으로 정의가 가능하다.
- CPN의 성질을 이용하여 시스템을 구현하기 이전에 안정성을 검증할 수 있다.

칼라 페트리 네트는 분산되어있고, 동시 발생적이며, 결정적이고, 동기화방식의 비결정적인 시스템들에 대하여 그래픽적인 모델링 언어로 표현이 가능하다. 그래픽적인 모델링 언어로 표현이 가능한 칼라 페트리 네트는 소스코드의 각 처리단계에 대하여 모든 가능한 상태의 각 단계를 발생 그래프로 표현이 가능하다.

칼라 페트리 네트로 안정성을 검증하는 방법에는 발생 그래프(Occurrence graph)와 상태 불변식(Place Invariant)이 있으나 본 논문에서는 발생그래프로 안정성을 검증하고자 한다. 안정성 검증을 실행하는 발생그래프(Occurrence-Graphs) 방법의 알고리즘의 구성은 다음 [그림 3]과 같다.

Waiting은 처리될 마킹들의 집합을 나타낸다. Waiting의 값을 초기화한 후에 노드를 M_0라고 설정한다. M_1을 선택한 후 모든 (b, M_2)에 대해 노드 M_1의 설정과 Arc(M_1, b, M_2)를 수행한다. Arc(M_1, b, M_2)는 M_1에서 M_2로 가는 바인딩 엘리먼트 b를 나타낸다. Waiting에서 M_1을 뺀 나머지를 Waiting으로 저장한다. Waiting의 값이 0이 될 때까지 반복해서 위의 단계들을 실행한다.

```
Waiting := 0
        Node(M0)
        repeat
                select a node M1 ∈ Waiting
                for all (b, M2) ∈ Next(M1) do
                begin
                        Node(M2)
                        Arc(M1, b, M2)
                end
                Waiting := Waiting - {M1}
        until Waiting = 0.
```

그림 3. 발생 그래프 알고리즘
Fig. 3. Algorithm of Occurrence graph.

발생 그래프(O-graphs)는 상태 공간(state spaces)이나 도달성 그래프(reachability graph)라고도 한다. 도달성 그래프에 대한 기본적인 개념은 도달 가능한 시스템의 상태를 나타내는 노드와 각각의 바인딩 구성요소(binding element)를 발생하기 위한 아크로 구성된다. 이러한 발생 그래프는 시스템의 변경 가능한 모든 상태를 나타내는 방향성 그래프를 구축하는 것으로, 이 그래프를 이용하여 모델의 분석 및 검증이 가능하다.

[그림 4]는 보안기능 일부를 칼라 페트리 네트의 발생그래프 방법으로 표현한 것으로 발생 그래프의 초기 단계에서 After_user는 Check_maclevel1의 규칙에 따라 다음 단계의 상태 After_maclevel1이 발생되고, After_maclevel1의 상태는 Check_maclevel1의 규칙이 만족한 경우 다음 규칙을 적용하기 위하여 Check_maclevel2로 상태가 전이되며, Check_maclevel1의 규칙이 만족하지 않는 경우는 로그기록 Process_logging1만 남기고 처리가 종료 eop가 된다. 만약 Check_maclevel2의 적용 규칙이 만족하는 경우는 Check_maclevel3가 발생되고 MAC의 처리단계는 끝나고 다음 단계를 수행하기 위한 After_MAC 상채가 발생하지만, 그렇지 않은 경우는 로그기록을 위한 Process_logging2만 남기고 처리가 종료 eop가 된다.
발생 그래프로 모델의 각 단계를 표현하였을 때, 처리과정이 무한 반복 상태(looping) 또는 처리 도중에 교착상태(deadlock)가 발생하지 않으면 안정성이 있음이 검증된다.

칼라 페트리 네트(Color Petri Nets) 프로그램에서 제공하는 도구는 다음과 같은 특징들을 가진다[2, 4, 8~9].

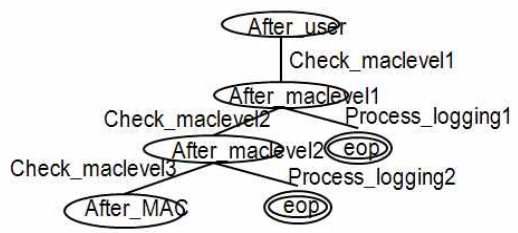

그림 4. 발생 그래프
Fig. 4. Occurrence graph.

- 칼라 페트리 네트를 생성, 수정 가능한 에디터제공.
- 칼라 페트리 네트를 검증하는 문장체크를 제공.
- 칼라 페트리 네트를 실행하는 시뮬레이터 제공.
- 대화형 모니터링과 디버깅 기능을 제공.
- 칼라 페트리 네트를 계층적 모듈 구성 기능 제공.
- 시뮬레이션 결과를 차트로 나타내는 기능 제공.

[표 1]은 [그림 4]의 발생그래프의 각 단계를 안정성 검증 테이블로 작성한 것이다.

표 1. 안정성 검증 테이블
Table 1. Secure Verification Table.

처음 \ 다음	Check_maclevel1	Check_maclevel2	Check_maclevel3	Process_logging1	Process_logging2
After_user	진행				
After_maclevel1		진행		eop	
After_maclevel2			진행		eop
After_MAC					

(1) CPN의 표현방법

CPN의 표현방법은 [그림 5]와 같으며, 칼라셋(color set)은 토큰의 타입을 나타낸다. 아크식은 아크의 표현값을 나타내고, 가드(Guard)식은 전이의 조건을 나타낸다. 초기 마킹은 플레이스에서 표현되어, 초기의 토큰 값을 나타내며, 선언노드는 현재 페이지에서 사용하는 칼라와 변수를 선언한다[4, 9, 11].

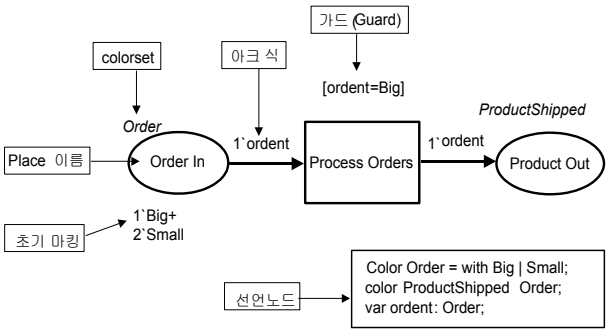

그림 5. CPN의 표현방법
Fig. 5. Express of CPNs.

나. 인증기능의 안정성 검증

사용자 인증기능 과정의 검증을 위한 표현은 [그림 6]과 같으며 인증을 체크하기 위한 토큰이 입력되면(check_auth request), 인증 플래그가 인증되지 않은 경우에만 명령어 인증과정(cmd_auth)이 실행되고, 만약

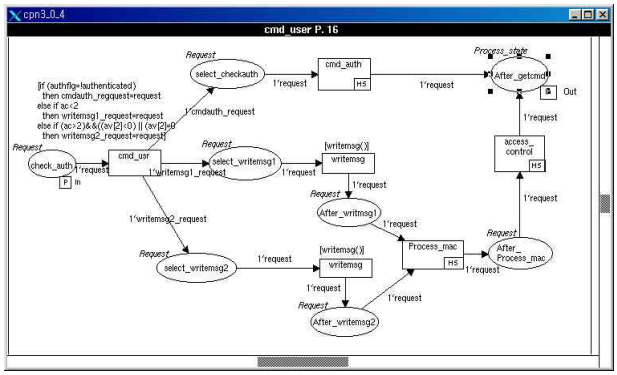

그림 6. 사용자 인증기능의 CPN 표현
Fig. 6. Design of CPN of User Authentication.

ac(argument count)가 2보다 작은 경우는 인증 요청(cmdauth_request)을 하고, ac가 2보다 크고 0보다 작거나 또는 0인 경우는 메시지를 남기는 작업을 위해서(writemsg2_request)를 요구한다.

다음으로 메시지를 남기는 작업(writemsg2_request)을 요구하는 경우는 기록(logging) 모듈이 처리된다.

CPN으로 표현된 사용자 인증기능의 안정성 검증을 위하여 각 처리과정을 발생그래프 방법으로 표현한 결과 발생 가능한 모든 상태들 중 반복(looping)되거나 중도에 중단(interrupt)상태가 발생하는 경우가 없기 때문에 사용자 인증처리과정의 각 단계의 안정성이 검증되었다[1, 4, 7].

[표 2]는 사용자 인증기능 처리의 각 단계에 대한 발생그래프 결과를 기반으로 테이블로 표현한 것이다. auth_failure, authsend_success, authrecv_success,

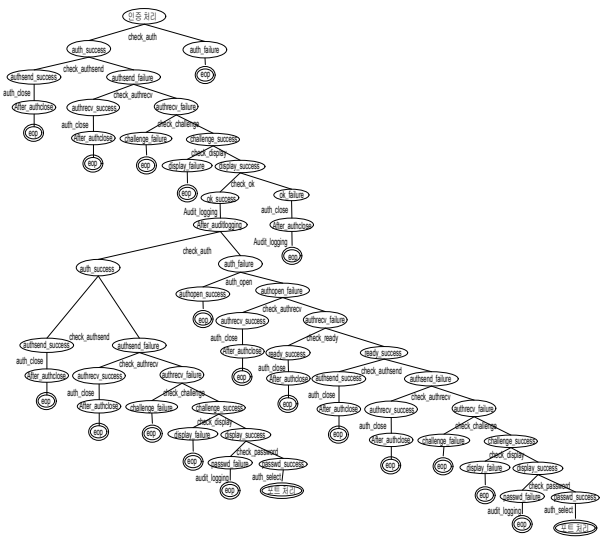

그림 7. 사용자 인증기능의 발생그래프
Fig. 7. Occurrence graph of User Authentication Processing.

표 2. 사용자 인증기능 안정성검증테이블(1단계)
Table 2. Secure Verification Table of User Authentication Processing (level 1).

처음 \ 다음	authsend_success	authsend_failure	authrecv_success	authrecv_failure	challenge_failure	challenge_success	display_failure	display_success	ok_success	ok_failure	최종상태
auth_success	진행	진행									
auth_failure											eop
authsend_success											eop
authsend_failure			진행	진행							
authrecv_success											eop
authrecv_failure					진행	진행					
challenge_failure											eop
challenge_success							진행	진행			
display_failure											eop
display_success									진행	진행	
ok_failure											eop

표 3. 사용자 인증기능 안정성 검증 테이블 (2단계)
Table 3. Secure Verification Table of User Authentication Processing (level 2).

처음 \ 다음	auth_success	auth_failure	authsend_succ	authsend_fail	authrecv_succ	authrecv_fail	challenge_failure	challenge_success	display_failure	display_success	password_fail	password_succ	최종상태
ok_success	진행	진행											
auth_success			진행	진행									
authsend_success					진행	진행							
authsend_failure													eop
authrecv_success													eop
authrecv_failure							진행	진행					
challenge_failure													eop
challenge_success									진행	진행			
display_failure													eop
display_success											진행	진행	
password_failure													eop
password_success													포트처리

challenge_failure, display_failure, ok_failure인 경우는 처리가 종료되어 최종 상태가 eop가 되며 나머지 처리 과정은 계속 진행 상태가 된다.

인증 처리의 각 단계의 안정성을 검증한 결과를 테이블의 계속 진행되는 상태가 [표 3]과 같으며, password_success인 경우만 “포트처리” 되고, authsend_failure, authrecv_success, challenge_failure, display_failure, password_failure인 경우는 처리가 종료되어 최종 상태가 eop가 되며 나머지 처리과정은 계속 진행 상태가 된다.

표 4. 사용자 인증기능 안정성검증테이블(3단계)
Table 4. Secure Verification Table of User Authentication Processing (level 3).

처음 \ 다음	authopen_succ	authopen_fail	authrecv_succ	authrecv_fail	challenge_fail	challenge_succ	display_fail	display_succ	password_fail	password_succ	최종상태
auth_failure	진행	진행									
authopen_success			진행	진행							
authopen_failure											eop
authrecv_success											eop
authrecv_failure					진행	진행					
ready_failure											eop
ready_success							진행	진행			
authsend_failure											eop
authsend_success									진행	진행	
challenge_failure											eop
challenge_success											포트처리
display_failure											eop
display_success											포트처리
password_failure											eop
password_success											포트처리

표 5. 사용자 인증기능 안정성검증테이블 (4단계)
Table 5. Secure Verification Table of User Authentication Processing (level 4).

처음 \ 다음	auth open_succ	auth open_fail	auth recv_succ	auth recv_fail	ready_fail	ready_succ	auth send_fail	auth send_succ	chall-enge_fail	chall-enge_succ	dis-play_fail	dis-play_succ	pas wo d_fa	pass word_succ	최종상태
auth_failure	진행	진행													
authopen_success															eop
authopen_failure			진행	진행											
authrecv_success															eop
authrecv_failure					진행	진행									
ready_failure															eop
ready_success							진행	진행							
authsend_failure									진행	진행					
authsend_success															eop
challenge_failure															eop
challenge_success											진행	진행			
display_failure															eop
display_success													진	진행	
password_failure															eop
password_success															포트처리

[표 4]는 challenge_success, display_success, password_success인 경우만 “포트처리” 로 진행이 되었다.

[표 5]는 password_success인 경우만 “포트처리” 로 진행이 되고, 인증과정이 완료된 후 마지막 포트처리는 다음 서버연결을 위한 단계로 진행이 된다.

Ⅲ. 결 론

네트워크와 시스템에 대해 적용한 침입방지시스템은 필수적인 보안 솔루션 중의 하나이며, 이러한 침입방지 시스템을 도입 시 가장 우선적으로 고려해야 될 사항으로는 다양한 기능보다 안정성이다. 이러한 보안 기능이 실행되기 위해서 입력된 데이터는 어떤 상황에서도 손실되지 않아야 하므로 실행과정이 무한 루프가 되어 계속 순환하거나, 교착 상태가 발생되지 않도록 시스템이 운영되기 위한 안정성을 검증하여야만 한다. 이러한 안정성 검증은 설계한 모델이 완벽하게 시스템에 구현되기 이전에 이루어지는 과정이다. 그러므로 본 논문은 침입방지시스템 보안기능 모델 중에서 사용자인증 과정에 대한 안정성을 검증하고자 컬러 페트리 네트를 이용하였다. 컬러 페트리 네트는 시스템 설계의 모델에 대한 각 단계를 그래픽적인 모델링 언어로 표현할 수 있고, 표현된 각 단계는 발생그래프를 이용하여 시스템의 처리과정이 무한 반복상태(looping) 또는 처리도중에 교착상태(deadlock)가 발생하지 않았음을 검증하게 되고 발생그래프로 검증된 내용은 처리의 상태와 규칙 그리고 발생 상태에 대한하여 테이블로 정리함으로써 최종적으로 안정성을 검증하였다. CPN을 이용한 이러한 검증과정은 차후에 침입방지시스템에서 제공되는 여러 보안기능의 설계 후 구현단계에 앞서 안정성 검증을 위한 유용한 도구이다. 차후 침입방지 시스템의 다른 보안기능과 처리과정 들에 대하여도 지속적인 개발과 안정성 검증을 하고자 한다.

참 고 문 헌

[1] Hui Kang, Xiuli Yang, Sinmiao Yuan, “Modeling and Verification of Web Services Composition based on CPN,” npc, pp.613-617, 2007 IFIP International Conference on Network and Parallel Computing Workshops (NPC 2007), 2007.

[2] Jaegeol Yim, Seunghwan Jeong, Jaehun Joo, “Colored Petri Net Representation of RDF Models,” fgcns, vol. 5, pp.46-51, 2008 Second International Conference on Future Generation Communication and Networking Symposia, 2008.

[3] Lian-zhang Zhu, Hua Zhang, “Queuing Network Models Analysis Based on CPN,” icic, vol. 2, pp.269-272, 2009 Second International Conference on Information and Computing Science, 2009.

[4] Vijay Gehlot, “Systems Modeling and Analysis Using Colored Petri Nets”, IEEE Systems Conference 2008.

[5] Xiang Wang, Yaxuan Qi, Baohua Yang, Yibo Xue, Jun Li, “Towards High-Performance Network Intrusion Prevention System on Multi-core Network Services Processor,” IEEE icpads, pp.220-227, 2009 15th International Conference on Parallel and Distributed Systems, 2009.

[6] Xinyou Zhang, Chengzhong Li, Wenbin Zheng, “Intrusion Prevention System Design,” cit, pp.386-390, Fourth International Conference on Computer and Information Technology (CIT'04), 2004.

[7] Yanxiang He; Tao Liu; Hai Zhong; Qi Xiong, “A CPN-Based Simulation Platform for Analysis and Defense Design of Internet End-Systems Targeted Attacks” FCST(Frontier of Computer Science and Technology) '09. Fourth International Conference, pp.548 - 552, 2009.

[8] Yongwei Wang; Shaowen Yao; Ying Zhao; Mingtian Zhou, “CPN modeling and analysis of L2TP” Computer Networks and Mobile Computing, pp.281 - 288, 2001.

[9] http://www.daimi.au.dk/CPnets/intro/example_indu.html

[10] 안정철 “ 침입방지시스템과 역할기반 보안정책을 이용한 정부기관 정보보호 시스템 설계” 한국정보보호학회, 정보보호학회논문지, 제14권 제6호 2004.12, page(s): 91-103

[11] 이문구 “침입차단 방화벽 시스템을 위한 FTP 프록시 모델의 설계 및 안정성 검증“ 숭실대학교 대학원, 컴퓨터학과, 박사학위논문, 1999. 12.

저 자 소 개

이 문 구(평생회원)

1984년 숭실대학교 전자계산학 (학사)

1993년 이화여자대학교 대학원 전산교육학 (석사)

2000년 숭실대학교 대학원 컴퓨터시스템 (공학 박사)

2000년 3월～현재 김포대학 IT 학부 인터넷정보과 부교수

<주관심분야 : 암호화 알고리즘, 인터넷 보안, 전자상거래 보안, 시스템 보안>