컬러드 페트리 네트를 기반으로 한 FTP 프록시 보안 모델의 안전성 검증

(Security Verification of FTP-Proxy Security Model Coloured Petri Net)

이 문 구 † 전 문 석 ††
(Moon-Ku Lee) (Moon-Seog Jun)

요 약 기존의 호스트 기반 보안 시스템에서 네트워크 기반의 보안 시스템으로 침입차단 시스템을 설치하지만, 침입차단 시스템은 보안에 대하여 최소 권한을 가지므로 사용자에게 투명한 서비스를 제공해 주지 못한다. 따라서 침입차단 시스템에 프록시를 두어서 서비스의 투명성을 제공하고, 보다 강화된 보안 기능을 갖는 FTP 프록시 보안 모델(FTP-PSM : FTP-Proxy Security Model)을 설계하였다. FTP-PSM은 한 개의 명령어가 입력되고 실행이 끝나는 것이 아니라 계속적으로 사용자 인증 기능, 강제적 접근제어 기능, 임의적 접근제어 기능 그리고 그룹별 명령어 사용권한 인증 등과 같은 보안 기능을 실행한다. 이러한 보안 기능이 실행되기 위해서 입력된 데이타는 어떤 상황에서도 손실되지 않아야 하므로 실행과정이 무한 루프가 되어 계속 순환하거나, 교착 상태가 되는 등의 문제에 대한 안전성을 검증하여야만 한다. 따라서 본 논문에서 제안하는 FTP-PSM은 CPN(Coloured Petri Net)을 기반으로 한 상태 불변식(Place Invariant)으로 안전성을 검증하였다.

Abstract The firewall systems can be installed between the internal network and the external network. The firewall systems has the least privilege, so its does not provide transparency to user. This problem of transparency can be solved by using the proxy. In this thesis, I have designed and verified the FTP-PSM(FTP-Proxy Security Model) which provides transparency for the firewall systems and has a strong security function. FTP-PSM doesn't finish its work after implementing a command. Instead, its does several security functions such as user authentication, MAC(Mandatory Access Control), DAC(Discretionary Access Control) and authentication of user group. Those data must not be lost under any circumstances in order to implement the above security functions. So, the security against such problems as falling into deadlock or unlimited loop during the implementation must be verified. Therefore, FTP-PSM suggested in thesis was verified its security through PI(Place Invariant) based on CPN(Coloured Petri Net).

1. 서 론

최근 인터넷에 대한 관심의 증대와 인터넷에 연결된 호스트의 숫자가 폭발적으로 증가함에 따라 사용자는 인터넷에 접속하여 다양한 형태의 정보 및 통신 서비스를 쉽게 제공받고 있다. 반면에, 인터넷을 통해 불법 사용자 및 해커의 침입 등으로 정보의 손실, 파괴, 변조 등에 의한 피해가 늘고 있다[2]. 인터넷으로부터 이러한 피해를 막기 위하여 침입 차단 시스템을 내부 네트워크와 외부 네트워크 사이에 설치하여 네트워크상의 한 지점에서 일괄적인 보안 정책을 적용시켜 관리함으로써 내부 네트워크의 정보 자산에 대하여 보안 수준을 높일 수 있다[3]. 그러나, 침입차단 시스템은 게이트웨이(gateway)역할을 하므로 네트워크 종단 사용자간의 인터페이스에서 투명한 서비스에 대하여 한계가 있다[4].

본 논문에서는 침입차단 시스템을 위하여 네트워크 인터페이스간의 투명한 서비스와 보안 기능을 갖는 FTP 프록시 보안 모델을 제안하고, 모델의 안전성 여부를 검증하기 위하여 CPN(Coloured Petri Net)의 상

· 이 논문은 2001학년도 김포대학의 연구비 지원에 의하여 연구되었음.
† 정 회 원 : 김포대학 인터넷정보전공 교수
yeon0330@kimpo.ac.kr
†† 종신회원 : 숭실대학교 컴퓨터학부 교수
mjun@computing.soongsil.ac.kr
논문접수 : 2000년 5월 26일
심사완료 : 2001년 8월 16일

태 불변식(Place Invariant)방법으로 검증하였다.

본 논문의 구성은 다음과 같다. 2장에서는 제안한 FTP 프록시 보안 모델을 설계하였고, 3장에서는 FTP 프록시 보안 모델(FTP-PSM : FTP-Proxy Security Model)의 안전성을 검증하기 위하여 도입한 CPN에 관한 이론을 기술한다. 그리고, 4장에서는 제안한 FTP-PSM을 CPN으로 표현한다. 5장에서는 상태 불변식(Place Invariant)방법으로 안전성을 검증한다. 그리고, 6장에서는 안전성이 검증된 FTP-PSM에 대한 결론과 차후 연구방향 등을 기술한다.

2. FTP-PSM (FTP-Proxy Security Model)의 설계

본 장에서는 보안 기능이 강화된 FTP 프록시 보안 모델(FTP-PSM)에서 제안하고자하는 보안기능을 설계하고 그 처리과정의 흐름도를 기술하였다.

2.1 FTP-PSM 의 보안기능 설계

FTP 프록시 보안 모델(FTP-PSM)은 침입 차단 시스템의 기능을 보다 효율적으로 하고 인터넷 보안 및 FTP의 보안 문제를 해결하기 위하여 다음과 같은 기능들을 설계하여 제공한다[그림 1].

- 일회용 패스워드를 이용한 사용자 인증 기능
- 강제적 접근 제어 기능
- 임의적 접근 제어 기능
- 사용자 그룹별 명령어 사용권한 인증 기능

FTP 서비스 요청이 들어오면 서비스 요청 처리 모듈에서 명령어들을 처리하기 위한 초기화과정이 이루어지고, 다음은 사용자의 인증 요청에 대하여 인증 처리과정을 갖게되어 사용자의 ID와 패스워드를 확인한다. FTP 프록시에서 사용되는 인증 방법으로는 일반적인 UNIX 시스템에서 사용되는 평문(plain text)인 단순한 패스워드 방식과 침입자들이 이용하고 있는 스니퍼(sniffer) 공격으로부터 근본적으로 막아주는 일회용 패스워드(One-Time Password)방법을 지원한다. 사용자 인증 과정이 끝나면, 주체 및 객체의 등급에 따라 강제적 접근제어를 실행하고 주체 및 객체의 신분에 근거한 임의적인 접근제어 과정이 이루어지고, 서버의 응답을 요구하는 모듈을 실행한다. 그리고 서버의 응답을 받으면 FTP 제어 명령어를 실행하기 위한 제어 연결이 완료되면 FTP의 자료를 전송하기 위하여 데이타 연결을 위한 설정과정을 갖는다. 이렇게 실제로 자료를 주고받는 처리과정을 하기 위해 FTP-PSM에서는 FTP 명령어의 사용권한이 인증된 사용자에게만 자료를 복사할 수 있도록 또 한번의 인증과정을 갖는다. 이러한 과정들은 각 모듈이 실행

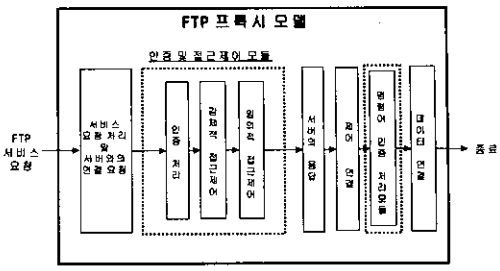

그림 1 제안한 FTP-PSM의 설계

될 때마다 로그 파일에 기록되므로 추후에 감사 자료가 될 수 있도록 한다.

FTP-PSM에서 보안 기능의 흐름도는 [그림 2]와 같으며, 그 실행 과정은 다음과 같다.

① 처음에는 인증이 안되어 있으므로 인증 여부를 0으로 설정하고, 명령어를 입력받은 후 서버와 연결이 되었는지 확인한다.

② 서버와 연결이 안되었으면 인증 여부가 1이 아니므로 사용자 인증 과정을 처리한 다음 인증 여부를 1로 설정하고 다시 명령어를 입력받는다.

③ 인증 여부가 1이면 즉, 인증을 거쳤으면 강제적 접근제어를 실행하고, 임의적 접근제어가 허용되면 서버와 연결이 된다.

④ 사용자 인증에서 거부되거나, 강제적 또는 임의적 접근제어가 실패했을 경우 감사기록을 남기고 인증 여부를 0으로 설정한 뒤 접속을 차단한다.

⑤ 서버와 연결이 되어있으므로 명령어를 체크하여 명령어 사용권한이 있는지를 확인하고, 명령어 사

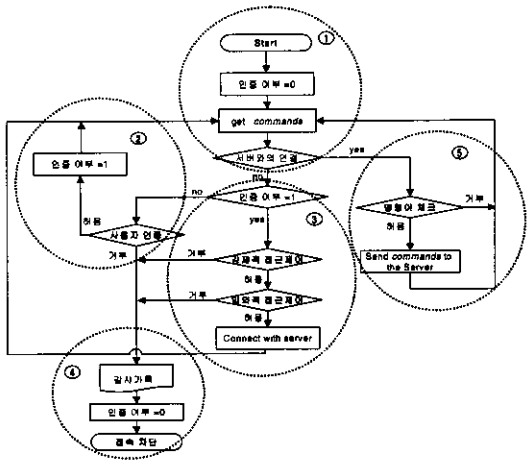

그림 2 FTP-PSM의 설계를 위한 보안 기능 흐름도

용권한이 확인되면 서버에게 바로 명령어를 실행하도록 보낸다. 만약, 명령어 사용에 대한 권한이 없으면 사용자에게 권한이 없음을 알리고 다음 명령어를 입력받는다.

3. 안전성 검증을 위한 CPN의 도입

본 장에서는 FTP-PSM의 각 기능들이 처리되는 과정에 대한 안전성 검증을 위하여 CPN을 도입하게된 목적과 CPN의 속성에 대하여 기술한다.

3.1 CPN 도입의 목적

설계한 FTP-PSM의 안전성을 검증하기 위하여 CPN (Coloured Petri Net)을 도입하게된 주요 목적은 다음과 같다[1].

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

CPN은 시스템의 각 기능에 대하여 동적 특성을 검증하기 위해서 사용된다. 이중에서도 특히 데드락 발생 여부, 토큰간의 충돌 발생 여부 등 안전성(safety)을 검증할 수 있고 형평성 등도 검증할 수 있다. 또한 CPN은 그 특성 때문에 상태 수 급증 문제 (state explosion problem)를 유발하지 않으면서 도달성 분석(reachability analysis)을 수행할 수 있기 때문에 여러가지 분석 방법을 적용할 수 있는데, 본 논문에서는 정형적인 방법으로 상태 불변식(Place Invariant)을 이용하여 제안하는 모델의 안전성을 검증하고자 한다.

3.2 CPN의 구성

CPN은 다음의 요구사항을 만족하는 튜플(tuple)로 구성된다.

[정의 3.1]

$CPN = (\sum, P, T, A, N, C, G, E, I)$

$\sum$: 0이 아닌 타입의 유한 집합 형태의 컬러 집합

$P=\{ P_1, P_2, \ldots, P_m\}$: 상태(Place)의 유한 집합으로 어떤 사건이 발생하기 전이나 발생한 후의 상태

$T=\{ t_1, t_2, \ldots, t_n\}$: 전이(Transition)의 유한 집합으로 어떤 상태로 도달하기 위한 사건

A : $A \subseteq (P \times T) \cup (T \times P)$를 나타내며, 흐름 관계의 유한 집합으로, 전이(transition)의 흐름

N : N은 노드(Node)의 함수 $N(a)$

(만약 source에서 destination으로 간다면, N=(source, dest)를 표현)

G : 가드(Guard)의 함수

$\forall_t \in T: [Type(G(t)) = B \wedge Type(Var(G(t))) \subseteq \sum]$

B는 바인딩 엘리먼트(binding element) b의 유한 집합

E : 간선 식 함수

C : 컬러(Color)의 함수

$\forall_a \in A : [Type(E(a)) = C(p(a))_{MS} \wedge Type(Var(E(a))) \subseteq \sum]$

$p(a)$는 $N(a)$의 상태(place)이다.

I : 초기화 함수로서, 다음의 닫힌 식 P 로부터 정의

$\forall_p \in P : [Type(I(p)) = C(p)_{MS}]$

가드 함수 G는 변환 t와 대수형의 식 즉, 술어를 사상한다. G(t)의 모든 변수는 $\sum$ 에 속하는 데이타 타입을 가져야만 한다. 모든 노드의 집합을 표시하기 위해서 $X = P \cup T$를 사용한다. 그리고 CPN 구조의 이웃 요소들 사이의 관계를 기술하는 많은 함수를 정의한다. 각 함수의 이름은 함수의 범위를 나타내며, P는 상태에 사상되고, A는 간선의 집합에 사상된다.

컬러집합은 CPN모델에서 사용되는 유형, 연산 함수들을 결정한다. 각 컬러집합은 적어도 한 개의 요소를 갖고 있다고 가정한다. 상태(P), 전이(T)와 간선(A)은 집합 P, T, A의 조합으로 정의된다. 상태, 전이, 그리고 간선을 갖는 집합은 유한해야 한다. 노드 함수는 첫 번 요소가 출발 노드이고, 두 번째가 목표 노드인 쌍을 각 간선으로 사상한다.

[정리 3.1]

W 가 상태(place)의 흐름을 나타내기 위한 필요충분 조건은 W 가 상태(place)의 불변식(invariant)을 결정하는 (i)과 (ii)를 만족하여야만 한다.

(i) 다음의 조건을 만족하는 필요충분 조건인 경우 W는 상태의 흐름(flow)이라고 말한다:

$$\forall (t, b) \in BE : \sum_{p \in P} W_p(E(p, t)\langle b \rangle) = \sum_{p \in P} W_p(E(t, p)\langle b \rangle). \quad (1)$$

W_p는 모든 $p \in P$ 에 대한 가중치의 집합이다.

BE 는 모든 바인딩 엘리먼트(binding element)들의 집합이고, 바인딩 엘리먼트는 (t, b)의 쌍으로 이루어지며, $t \in T$ 그리고 $b \in B(t)$이다.

(ii) 다음의 조건을 만족하는 필요충분 조건인 경우 W는 상태 불변식(place invariant)을 결정한다고 말한다:

$$\forall M \in [M_0\rangle : \sum_{p \in P} W_p(M(p)) = \sum_{p \in P} W_p(M_0(p)). \quad (2)$$

W가 상태(place)의 흐름(flow)이라고 가정할 때, $M_1[Y\rangle M_2$가 $W(M_1) = W(M_2)$임을 증명한다면 제안하는 모델의 안전성을 검증하게된다.

[증명]

Y는 초기 마킹 M_1에서 진행되는 단계를 표현한다. 그리고 Y 단계는

$\forall p \in P : \sum_{(t,b)\in Y} E(p,t)\langle b\rangle \le M(p)$의 필요충분 조건을 만족하는 마킹 M 에서 진행가능 하다. 마킹 M_1에서 진행 가능한 Y단계가 발생하면 마킹 M_1은 다른 마킹 M_2로 변경되며, 다음과 같이 표현된다:

$$\forall p \in P:\ M_2(p) = (M_1(p) - \sum_{(t,b)\in Y} E(p,t)\langle b\rangle) + \sum_{(t,b)\in Y} E(t,p)\langle b\rangle. \quad ①$$

첫 번째 합은 제거된 토큰들을 나타내고, 두 번째 합은 추가된 토큰들을 나타낸다. Y단계의 발생에 의해서 M_2는 M_1에서 직접 도달가능하며, $M_1[Y\rangle M_2$ 같이 표현된다.

식 ①에 의해서 [정리 4.1]의 식 (1)을 적용하면 식 ②와 같다:

$$\sum_{p\in P} W_p(M_2(p) + \sum_{(t,b)\in Y} E(p,t)\langle b\rangle) = \sum_{p\in P} W_p(M_1(p) + \sum_{(t,b)\in Y} E(t,p)\langle b\rangle). \quad ②$$

가중치 함수(weight function)의 선형성(linearity)으로부터 식 ③을 얻을 수 있다:

$$\sum_{p\in P} W_p(M_2(p)) + \sum_{p\in P}\sum_{(t,b)\in Y} W_p(E(p,t)\langle b\rangle) = \sum_{p\in P} W_p(M_1(p)) + \sum_{p\in P}\sum_{(t,b)\in Y} W_p(E(t,p)\langle b\rangle). \quad ③$$

흐름 성질(flow property)로부터 식 ④를 얻을 수 있다:

$$\forall (t,b)\in BE:\ \sum_{p\in P} W_p(E(p,t)\langle b\rangle) = \sum_{p\in P} W_p(E(t,p)\langle b\rangle). \quad ④$$

식 ④는 식 ⑤와 같이 적용된다:

$$\sum_{(t,b)\in Y}\sum_{p\in P} W_p(E(p,t)\langle b\rangle) = \sum_{(t,b)\in Y}\sum_{p\in P} W_p(E(t,p)\langle b\rangle). \quad ⑤$$

식 ⑤는 식 ⑥과 같이 다시 작성할 수 있다:

$$\sum_{p\in P}\sum_{(t,b)\in Y} W_p(E(p,t)\langle b\rangle) = \sum_{p\in P}\sum_{(t,b)\in Y} W_p(E(t,p)\langle b\rangle). \quad ⑥$$

위의 식에서 두 개의 $\sum$는 위의 식과 동일하기 때문에 식 ⑦과 같은 결론을 얻을 수 있다:

$$\sum_{p\in P} W_p(M_2(p)) = \sum_{p\in P} W_p(M_1(p))$$

즉, $W(M_2) = W(M_1)$이다. ⑦

다음에 $M\in[M_0\rangle$을 도달 가능한 마킹(marking)이라고 하고, σ를 M_0에서 시작해서 M 으로 끝나는 발생순서라고 하자. 위의 결과를 σ의 $M_i[Y_i\rangle[M_{i+1}$의 각 단계에 적용하면 $W(M) = W(M_0)$이라는 결론을 얻을 수 있다. 그러므로 [정리 3.1]의 (i)이 증명된다.

이번에는 [정리 3.1]의 (ii)를 증명하기 위해서 W 의 상태 불변식을 결정하고 CPN이 동작하지 않는 바인딩 구성요소는 가지고 있지 않다고 가정하자. 이것은 각 바인딩 구성요소 (t,b)가 적어도 도달가능한 하나의 M_1을 가지고 있다는 것을 의미한다. M_2를 $M_1[t,b\rangle M_2$에 의해 결정되는 마킹이라고 하자.

앞과 유사한 순서에 의해 $W(M_2) = W(M_1)$이 식 ⑧과 같이 적용됨을 알 수 있다:

$$\sum_{p\in P} W_p(E(p,t)\langle b\rangle) = \sum_{p\in P} W_p(E(t,p)\langle b\rangle). \quad ⑧$$

그러므로 [정리 3.1]의 (ii)가 증명됨을 알 수 있다.

4. FTP-PSM의 CPN 표현

본 장에서는 FTP-PSM에서 제안하는 각 보안기능들을 CPN으로 표현하였다.

4.1 FTP-PSM에서 제안하는 보안기능의 CPN 표현

CPN의 표현방법은 [그림 3]과 같으며, 칼라셋(color set)은 토큰의 타입을 나타낸다. 아크식은 아크의 표현값을 나타내고, 가드(Guard)식은 전이의 조건을 나타낸다. 초기 마킹은 플레이스에서 표현되어, 초기의 토큰값을 나타내며, 선언노드는 현재 페이지에서 사용하는 칼라와 변수를 선언한다.

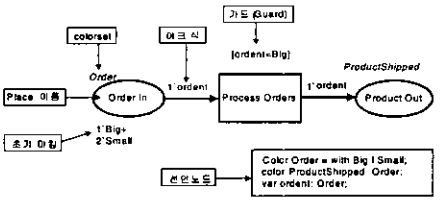

그림 3 CPN의 표현방법

FTP-PSM에서 제안하는 보안기능들은 CPN을 기반으로 다음[그림 4], [그림 5], [그림 6], [그림 7]과 같이 표현할 수 있다[5],[6],[7],[8].

4.1.1 사용자 인증기능 처리과정의 CPN 표현

사용자의 ID와 패스워드로 일반적인 패스워드 기능과 일회용 패스워드 기능을 사용하여 인증된 사용자인지를 검증하기 위하여 설계된 인증 기능을 CPN으로 표현하면 다음과 같다.

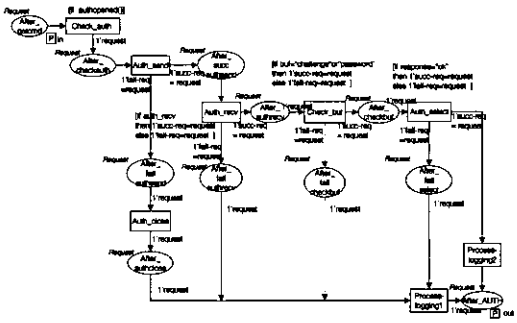

그림 4 사용자 인증기능 처리과정

4.1.2 임의적 접근제어 기능 처리과정의 CPN 표현

임의적 접근제어(DAC : Discretionary Access Control)는 주체의 신분에 근거한 보안 기능으로서, 주체에 대한 접근권한을 확인하여 객체에 대한 접근제어를 수행하도록 설계되며 처리과정은 다음과 같이 CPN으로 표현할 수 있다.

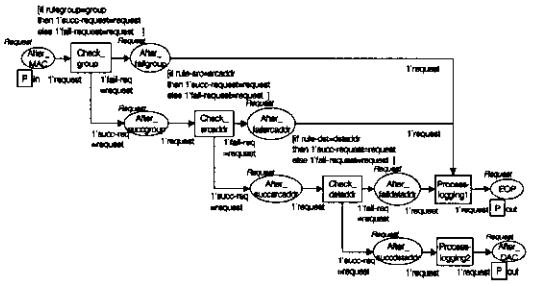

그림 5 임의적 접근제어(DAC) 처리과정

4.1.3 강제적 접근제어 기능 처리과정의 CPN 표현

강제적 접근제어(MAC : Mandatory Access Control)는 주체 및 객체의 보안 등급에 근거하여 주체의 객체에 대한 접근을 제어하는 방법으로, 주체 및 객체의 보안 등급에 따라 접근제어를 하므로, 임의적 접근제어에 비해 세밀한 접근제어가 가능하여 보안에 대한 높은 신뢰성을 제공한다. 설계된 강제적 접근제어 기능의 CPN 표현은 다음과 같다.

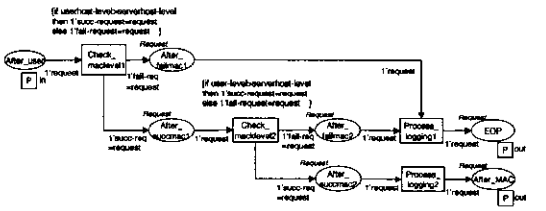

그림 6 강제적 접근제어(MAC) 처리과정

4.1.4 명령어 사용권한 인증기능 처리과정의 CPN 표현

서버와 연결된 후에 FTP-PSM에서는 사용자 그룹별로 FTP 명령어에 대한 사용권한을 제어한다. 따라서 초기 침입차단 시스템에 접속하고자 할 때 관리자에게 일부 FTP 명령어(예, CWD, REST, STOR, NLIST)에 대한 사용권한이 있는 사용자에 대해서만 실행이 가능하도록 하는 제어기능을 설계하였으며, 다음은 그 처리과정을 CPN으로 표현한 것이다.

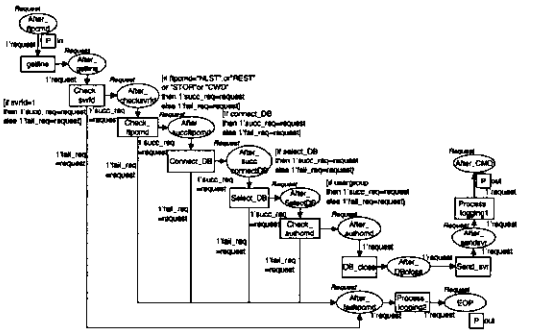

그림 7 명령어 사용권한 인증기능 처리과정

5. FTP-PSM의 안전성 검증

본 장에서는 FTP-PSM의 안전성 검증의 목적과 모델에서 설계된 보안기능 중에서 강제적 접근제어 기능에 대한 안전성 검증의 결과를 기술하였다.

5.1 FTP-PSM의 안전성 검증의 목적

FTP-PSM은 한 개의 명령어가 입력되고 실행이 끝나는 것이 아니라 계속적으로 사용자의 ID 및 IP 주소 그리고 FTP 명령어가 입력되어 사용자 인증 기능, 강제적 접근제어 기능, 임의적 접근제어 기능 그리고 그룹별 명령어 사용권한 인증 등과 같은 보안 기능이 실행된다. 이러한 보안 기능이 실행되기 위해서 입력된 데이타는 어떤 상황에서도 손실되지 않아야 하므로 실행과정이 계속 반복 순환하거나, 교착 상태가 발생하는지, 또는 데이타 충돌이 발생되는지 등에 관한 안전성을 검증하여야만 한다. 그리고 이러한 안전성은 설계한 모델이 완벽하게 시스템에 구현되기 위해서 가장 중요한 요인이다. 따라서 본 장에서는 FTP 프록시 모델의 안전성 여부를 검증하기 위하여 CPN(Coloured Petri Net)의 상태 불변식(Place Invariant) 검증 방법으로 FTP-PSM의 안전성을 검증한다.

상태 불변식은 CPN 성질의 안전성, 유계성, 보존성 등을 만족하며, 발생가능한 상태(place)와 전이의 토큰 개수가 유한적으로 정의됨으로써 모델에서 오버플로우(overflow)나 데드락(deadlock) 그리고 무한 루프(loop)가 발생하지 않음을 검증할 수 있다. 이러한 상태 불변식은 제안한 모델의 발생 가능한 모든 경우를 CPN으로 표현하고, CPN으로 표현된 각 보안기능 처리과정을 상태 불변식 검증 방법으로 실행하게 된다. 상태 불변식 검증 방법은 모델에서 표현되는 모든 상태를 방정식으로 표현하여 입력이 요청된 토큰과 출력되는 토큰이 항상 같도록 유지되어 상태가 불변한다는 정리를 수학적

으로 증명함으로써 모델의 안전성을 검증하게 된다[10].

5.2 강제적 접근제어 기능의 안전성 검증

본 장에서는 FTP-PSM의 보안기능 중에서 강제적 접근제어 기능에 대하여 상태 불변식으로 그 안전성을 검증하고자 한다.

FTP-SPM의 안전성을 검증하기 위해서 강제적 접근제어 기능의 처리과정은 다음과 같이 3가지 도달 가능한 상태로 표현 할 수 있다.

① $M_{After-user}[\ Y_{Chek-macleve1},\ Y_{Process-logging}\rangle\ M_{EOP}$

② $M_{After-user}[\ \begin{matrix} Y_{Chek-macleve1},\ Y_{Chek-macleve2} \\ Y_{Process-logging1}\rangle \end{matrix}\ M_{EOP}$

③ $M_{After-user}[\ \begin{matrix} Y_{Chek-macleve1},\ Y_{Chek-macleve2} \\ Y_{Process-logging2}\rangle \end{matrix}\ M_{After-MAC}$

첫 번째 도달 가능한 상태 ①의

$M_{After-user}[\ Y_{Chek-macleve1},\ Y_{Process-logging}\rangle\ M_{EOP}$은 식 (a)와 같이 표현 할 수 있다.

$$M_{After-user}\ \begin{matrix} [\ Y_{Chek-macleve1}\rangle\ M_{After-failmac1} \\ [\ Y_{Process-logging}\rangle\ M_{EOP} \end{matrix} \qquad (a)$$

식(a)에서 $M_{After-user}[\ Y_{Chek-macleve1}\rangle\ M_{After-failmac1}$

이 $W(M_{After-user}) = W(M_{After-failmac1})$ 와

$\sum_{p\in P} W_p(E(p,t)\langle request\rangle) = \sum_{p\in P} W_p(E(t,p)\langle request\rangle)$

임을 증명하면 FTP-PSM에서 강제적 접근제어 기능이 안전성이 있음을 검증하게 된다.

(a)는 [정리 3.1]에 따라 식 ①'를 얻을 수 있다.

$$\forall p\in P:\ M_2(After-failmac1) = (M_1(After-user) - \sum_{(t,b)\in Y} E(p,t)\langle request\rangle) + \sum_{(t,b)\in Y} E(t,p)\langle request\rangle. \qquad ①'$$

첫 번째 합은 제거된 토큰들을 나타내고, 두 번째 합은 추가된 토큰들을 나타낸다.

Y단계의 발생에 의해서 M_2는 M_1 에서 직접 도달가능하며,

식 ①'에 [정리 3.1]의 (i)를 적용하면 식 ②'과 같다.

$$\sum_{p\in P} W_p(\ M_2(After-failmac1) + \sum_{(t,b)\in Y} E(p,t)\langle request\rangle) = \sum_{p\in P} W_p(M_1(After-user) + \sum_{(t,b)\in Y} E(t,p)\langle request\rangle) \qquad ②'$$

식 ②'은 가중치 함수(weight function)의 선형성(linearity)으로부터 다음 식 ③' 을 얻을 수 있다.

$$\sum_{p\in P} W_p(\ M_2(After-failkmac1)) + \sum_{p\in P}\sum_{(t,b)\in Y} W_p(E(p,t)\langle request\rangle) = \sum_{p\in P} W_p(M_1(After-user)) + \sum_{p\in P}\sum_{(t,b)\in Y} W_p(E(t,p)\langle request\rangle) \qquad ③'$$

식 ③'은 흐름 성질(flow property)로부터 다음 식 ④'을 얻을 수 있다.

$$\forall(t,b)\in BE:\ \sum_{p\in P} W_p(E(p,t)\langle request\rangle) = \sum_{p\in P} W_p(E(t,p)\langle request\rangle) \qquad ④'$$

식 ④'는 식 ⑤'과 같이 적용된다.

$$\sum_{(t,b)\in Y}\sum_{p\in P} W_p(E(p,t)\langle request\rangle) = \sum_{(t,b)\in Y}\sum_{p\in P} W_p(E(t,p)\langle request\rangle) \qquad ⑤'$$

식 ⑤'는 다음 식 ⑥'과 같이 다시 작성할 수 있다.

$$\sum_{p\in P}\sum_{(t,b)\in Y} W_p(E(p,t)\langle request\rangle) = \sum_{p\in P}\sum_{(t,b)\in Y} W_p(E(t,p)\langle request\rangle) \qquad ⑥'$$

위의 식 ⑥'에서 두 개의 $\sum$는 위의 식과 동일하기 때문에 다음 식 ⑦'과 같은 결론을 얻을 수 있다.

$$\sum_{p\in P} W_p(M_2(After-failmac1)) = \sum_{p\in P} W_p(M_1(After-user)).$$

즉, $W(M_{After-failmac1}) = W(M_{After-user})$이다. ⑦'

다음에 $M_{EOP}\in[\ M_{After-user}\rangle$을 도달가능한 마킹(marking)이라고 하고, σ를 $M_{After-user}$에서 시작해서 M_{EOP} 으로 끝나는 발생순서라고 하자.

위의 결과를 σ의 $M_{After-user}[\ Y_{Chek-macleve1}\ Y_{Process-logging}\rangle M_{EOP}$의 각 단계에 적용하면 $W(M_{EOP}) = W(M_{After-user})$이라는 결론을 얻을 수 있다.

그러므로 [정리 3.1]의 (i)이 증명된다.

이번에는 $\sum_{p\in P} W_p(E(p,t)\langle b\rangle) = \sum_{p\in P} W_p(E(t,p)\langle b\rangle)$를 증명하기 위해서 W의 상태 불변식을 결정하고 FTP- PSM은 CPN이 동작하지 않는 바인딩 구성 엘리먼트는 가지고 있지 않다고 하자. 이것은 각 바인딩 엘리먼트 (t,b)가 적어도 도달가능한 하나의 M_1을 가지고 있다는 것을 의미한다. $M_{After-failmac1}$을 $M_{After-user}[t,b\rangle M_{After-failmac1}$에 의해 결정되는 마킹이라고 하자.

여기서 t와 b를 다시 정의하면, t는 전이(transition)의 엘리먼트(element)이고, b는 바인딩(binding) 엘리먼트이다.

즉, [그림 6]에서 첫 번째 발생가능한 단계 $Y_{Chek-macleve1}$의 가드 식 $G(t)$ 는 $G(t)$=(userhost-level>serverhost-level)이며, 사용자의 호스트 등급과 서버의 호스트 등급에 따라서 바인딩 엘리먼트<b>값이 succ-request 또는 fail-request로 결정된다. 단 succ-request와 fail-request 그리고 request는 동일한 request의 컬러 타입을 갖는다.

따라서, $M_{After-user}[t, b\rangle M_{After-failmac1}$ 에서 t 의 조건식에서 사용자 호스트 등급이 서버의 호스트 등급보다 낮다면 b 는 fail-request로서 $M_{After-failmac1}$상태의 마킹 값을 갖게된다. 그러므로, $M_{After-failmac1}$는 $M_{After-user}[t, b\rangle M_{After-failmac1}$에 의해 결정되는 마킹이므로, 앞서 증명된 결과에 따라서 $W(M_{After-failmac1}) = W(M_{After-user})$가 [정리 3.1]의 (ii)에 따라 다음 식(2)에 적용되어,

$$\forall M \in [M_0\rangle : \sum_{p\in P} W_p(M(p)) = \sum_{p\in P} W_p(M_0(p)). \quad (2)$$

$$\sum_{p\in P} W_p(E(p, t)\langle request\rangle) = \sum_{p\in P} W_p(E(t, p)\langle request\rangle) \quad ⑧'$$

⑧'와 같이 증명된다.

또한 CPN의 속성 중에서 보존성 성질에서 전이(transition)의 입력 토큰의 수는 출력 토큰의 수와 같아야 한다. 즉, $|I(t_i)| = |O(t_i)|$이므로 FTP-PSM의 강제적 접근제어 기능에서 전이(transition)의 조건식 $G(t)$에 따라 입력 토큰의 개수와 출력 토큰의 개수가 같으므로 전이 불변식(transition invariant)이 상태 불변식과 같은 방법으로 증명될 수 있으므로 FTP-PSM의 강제적 접근제어 기능은 안전성이 있음이 증명된다.

강제적 접근제어 기능의 각 도달 가능한 상태들은 다시 세부적으로 다음과 같이 정의 할 수 있다.

① $M_{After-user}[\ Y_{Chek-maclevel1}\ Y_{Process-logging}\rangle\ M_{EOP}$

$M_{After-user}[\ Y_{Chek-maclevel1}\rangle\ M_{failmac1}$

$M_{failmac1}[\ Y_{Process-logging}\rangle\ M_{EOP}$

② $M_{After-user}[\ Y_{Chek-maclevel1}\ Y_{Chek-maclevel2}\ Y_{Process-logging1}\rangle\ M_{EOP}$

$M_{After-user}[\ Y_{Chek-maclevel1}\rangle\ M_{After-sucmac1}$

$M_{After-sucmac1}[\ Y_{Chek-maclevel2}\rangle\ M_{After-failmac2}$

$M_{After-failmac2}[\ Y_{Process-logging1}\rangle\ M_{EOP}$

③ $M_{After-user}[\ Y_{Chek-maclevel1}\ Y_{Chek-maclevel2}\ Y_{Process-logging2}\rangle\ M_{After-MAC}$

$M_{After-user}[\ Y_{Chek-maclevel1}\rangle\ M_{After-sucmac1}$

$M_{After-sucmac1}[\ Y_{Chek-maclevel2}\rangle\ M_{After-sucmac2}$

$M_{After-sucmac2}[\ Y_{Process-logging2}\rangle\ M_{After-MAC}$

CPN으로 표현한 강제적 접근제어기능에서 각각의 발생 가능한 상태(place)들은 앞서 실행한 방법과 같은 방법으로 안전성을 검증할 수 있을 뿐만 아니라 FTP-PSM의 CPN으로 표현된 사용자 인증 기능[그림 4], 임의적 접근제어 기능[그림 5], 명령어 사용권한 인증기능[그림 7]들도 같은 방법으로 그 안전성을 검증할 수 있다.

6. 결 론

인터넷 사용이 급성장하고 많은 학교나 기업체 등의 네트워크가 인터넷을 통해 공유되면서 누구에게나 접속이 허용되어 많은 보안상의 문제가 발생하게 되었다. 이러한 문제들을 해결하기 위해 보안 기능이 강화된 FTP 프록시가 필요하다. 프록시는 정당한 사용자만이 정당한 서비스를 받을 수 있게 하기 위하여 인증 기능을 제공한다. 인터넷을 통해 간단한 인증 기능이 수행될 때 사용자의 ID와 패스워드는 인터넷상에서 암호화되지 않은 채로 전송되기 때문에 인터넷상의 패킷을 가로채는 프로그램을 이용할 경우에는 사용자의 ID와 패스워드가 공개될 수 있다. 이러한 문제를 해결하기 위해 사용하는 일회용 패스워드는 매번 인증할 때마다 사용하는 패스워드가 다르기 때문에 패스워드가 공개되어도 시스템을 보호할 수 있다.

침입 차단 시스템이 내부망을 보호하기 위하여 내부망과 외부망을 무조건 차단한다면 내부망과 외부망간의 융통성 있는 서비스가 제공되지 않는다. 또한 접근이 허가된 사용자인지를 정확하게 제어하여야만 한다. 이러한 접근제어 기능은 불법 사용자가 도중에 정보를 가로채어 정보의 비밀성을 침해하는 행위, 불법으로 접근하여 데이타를 변경하여 데이타의 무결성 손상, 혹은 허가되지 않은 주체가 시스템에 거짓 정보를 삽입하는 등의 보안 위협으로부터 공격을 받을 수 있다. 그러므로 본 논문에서 제안하는 FTP 프록시 보안 모델에서는 이처럼 내부망과 외부망 사이의 원활한 서비스 제공을 위한 투명성을 제공하면서 접근제어를 위하여 임의적 접근제어와 강제적 접근제어 기능 등을 제공하도록 하였다. 임의적 접근제어 기능은 사용자의 IP 주소에 따라 접속을 허용하거나 거부하는 방법으로 주체 및 객체의 신분에 따라 그 접근제어가 이루어진다. 그러나 임의적 접근제어 기능은 IP 주소에 따라 서비스를 허용하거나 제한하기 때문에 특정 사이트의 사용자 등급에 따라 서비스를 허용하거나 제한할 수 없다. 이러한 한계를 해결하기 위해 사용자의 등급과 접속하고자 하는 객체에 따라 서비스를 허용하거나 제한하는 강제적 접근제어 기능이 필요하다.

이렇게 접근이 허가된 사용자에 대해서도 FTP 명령어 사용에 대한 권한을 관리자로부터 부여받지 않은 사용자를 확인하는 인증 과정을 갖는다. 또한 접근이 이루어진 후에 사용자의 활동 상황을 감시하고, 내부통제 및 감사를 위한 로그기록 과정이 필요하다.

본 논문에서는 침입 차단 시스템을 위한 FTP 프록시

보안 모델(FTP-PSM)을 설계하고 실제로 침입 차단 시스템에 구현하기 이전에 모델에서 설계된 보안 기능들에 대한 안전성을 검증하고자 CPN 기반의 상태 불변식을 이용하였다. 상태 불변식은 CPN을 이용해서 시스템의 각 기능에 대한 흐름을 그래픽으로 표현이 가능하고, 그래픽으로 표현된 모든 가능한 상태를 방정식으로 표현하여 입력이 요청된 토큰과 출력되는 토큰이 항상 같도록 유지되어 상태가 불변한다는 정리를 수학적으로 증명함으로써 모델의 안전성을 검증하였다. 이러한 CPN 기반의 상태 불변식을 이용한 안전성 검증은 모델의 설계뿐만 아니라 보다 효율적인 구현단계로 실현될 수 있는 검증방법으로, 차후 보안 시스템 모델 설계에 대하여 안전성을 검증할 수 있는 기반이 될 수 있다.

참 고 문 헌

[1] Analysis of CP-nets, *http://www.daimi.aau.edu/CPnets/intro/analysis.htm*, December 1997.

[2] Bill Cheswick and Steve Bellovin, *"Firewalls and Internet Security: Repelling the Wily Hacker,"* Addison Wesley Edition, 1994.

[3] Chris Hare & Karanjit Siyan, *"Internet Firewall and Network Security"* Second Edition, New Riders, 1996.

[4] Douglas Comer and David Stevens, *"Internetworking with TCP/IP Vols I, II and III,"* Prentice-Hall. 1991.

[5] Design/CPN Overview of CPN ML Syntax, Meta Software, 1993.

[6] Design/CPN Programmer's Manual for X-Windows Version 2.0, Meta Software, 1993.

[7] Design/CPN Reference Manual for X-Windows Version 2.0, Meta Software, 1993.

[8] Design/CPN Occurence Graph Manual for X-Windows Version 2.0, Meta Software, 1993.

[9] Karanjit Siyank & Chris Hare, *"Internet Firewalls and Network Security,"* New Riders Publishing, 1995.

[10] Kurt Jensen, *"Colored Petri Nets. Basic Concepts, Analysis Methods and Practical Use"*. Volume 1, 2, 3: Basic Concepts, EATCS monographs on Theoretical Computer Science, Springer-Verlag 1992.

이 문 구

1984년 숭실대학교 전산과(학사). 1993년 이화여자대학교 교육대학원 전산학과(석사). 2000년 숭실대학교 대학원 전산과(공학 박사). 1997년 ~ 2000년 2월 명지 전문대학 컴퓨터과 경임교수. 2000년 2월 ~ 2001년 현재 김포대학 컴퓨터계열 인터넷정보 전임강사. 관심분야는 네트워크프로그램, 네트워크보안, 인터넷보안, 암호알고리즘

전 문 석

1980년 숭실대학교 전자계산학과(학사). 1986년 University of Maryland, Computer Science(석사). 1988년 University of Maryland, Computer Science(박사). 1989년 Morgan State Univ. 부설 Physical Science Lab. 연구원. 1991년 ~ 현재 숭실대학교 컴퓨터학부 부교수. 관심분야는 병렬처리 시스템, 침입차단 시스템, 암호알고리즘