2001년 한국정보처리학회 추계 학술발표논문집 제8권 제2호

네트워크 보안 정책 정보 모델을 위한 패킷 헤더 필드의 변수 정의

김건량, 김숙연, 김기영, 장종수
한국전자통신연구원

e-mail:{glkim, sykim, kykim, jsjang}@etri.re.kr

The Variable Definition of Packet Header Fields for Network Security Policy Information Model

Geon-Lyang Kim, Sook-Yeon Kim, Ki-Young Kim, Jong-Su Jang
Electronics and Telecommunications Research Institute

요 약

침입 및 해킹의 사례가 증가함에 따라 네트워크 침입 탐지 및 해킹 대응을 위한 네트워크 보안의 필요성이 증가하고 있으며, 정책 서버를 구축하는 솔루션이 등장하고 있다. 일반적인 정책 정보 모델은 IETF의 정책 프레임워크 워킹 그룹과 DMTF의 CIM 활동을 통해 활발히 표준화가 되고 있다. 이러한 표준들은 그 동안 QoS를 위해 대부분 사용되었으나 우리는 이러한 표준을 네트워크 보안 정책 시스템에 맞게 확장하여 네트워크 보안 정책 정보 모델을 구축한다. 본 논문은 패킷 헤더 필드들을 변수화하고 네트워크 보안 정책 정보 모델에서 침입 탐지 및 해킹 대응에 대한 정책을 모델링하는 방법을 제시한다.

1. 서론

최근 인터넷의 활성화로 인해 네트워크를 접하는 사용자가 증가하고 있으며, 침입 및 해킹의 사례 또한 증가하고 있다. 이에 따라 네트워크 침입 탐지 및 해킹 대응을 위한 네트워크 보안이 필요한데, 네트워크 보안을 위해 정책 서버를 구축하는 솔루션이 등장하고 있다.[KYKIM][SYKIM]

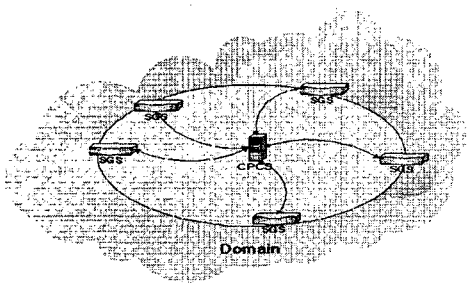
정책 서버에 기반한 네트워크 보안 시스템은 모든 정책을 관리하고 정책을 내리는 정책 서버와 이러한 정책을 수행하여 결과를 다시 정책 서버에 반환하는 여러 클라이언트들로 구성되어 있다. 이러한 시스템은 정책 정보 모델의 구축이 필수적이다.

일반적인 정책 정보 모델은 IETF의 정책 프레임워크 워킹 그룹과 DMTF의 CIM(Common Information Model)활동을 통해 활발히 표준화가 되고 있다. 그리고, PCIM(Policy Core Information Management)이 최초로 RFC3060으로써 표준화되었으며, 이 표준의 확장도 진행되고 있다. PCIM은 정책 정보와 제어를 표현하는 구조적인 클래스와 구조적인 클래스들의 인스턴스들이 서로 어떻게 연관되는지를 지시하는 연관 클래스들을 정의하고 있다. 이 정책 정보 모델은 핵심 모델로써 QoS와 IPSet과 같이 어플리케이션과 관련된 정책들을 표현하기 위해서 PCIM을 확장하여 사용하면 된다. 이러한 표준들은 그 동안 QoS 등을 위한 정책에 사용되었지만 본 논문에서 소개하는 정책 정보 모델은 네트워크 보안에 사용되는 정책을 위한 모델이다. 본 논문은 IETF 정책 프레임워크 워킹 그룹의 PCIM을 네트워크 보안 정책에 맞게 확장하여 침입 탐지 및 대응에 대한 정책을 모델링하는데, 패킷 헤더 필드들을 변수화하여 모델링에 사용한다.

본 논문은 1장 서론을 시작으로, 2장에서는 정책 기반의 네트워크 보안 시스템의 내부 구조를, 3장에서는 패킷 헤더 필드들을 변수로 정의하는 것과 이를 기반으로 침입 탐지 및 대응 정책의 모델링 과정을 설명하고, 4장에서 결론으로 마무리짓는다.

2. 정책 기반의 네트워크 보안 시스템의 구조

본 논문이 제안하는 정책 기반의 네트워크 보안 시스템은 아래 [그림 1]과 같이 하나의 사이버 순찰 제어 시스템(Cyber Patrol Control System-CPCS)과 여러 대의 보안 게이트웨이 시스템(Security Gateway System-SGS)으로 하나의 도메인을 구성하며, 여러 도메인을 관리하는 시스템으로 확장될 수 있다.



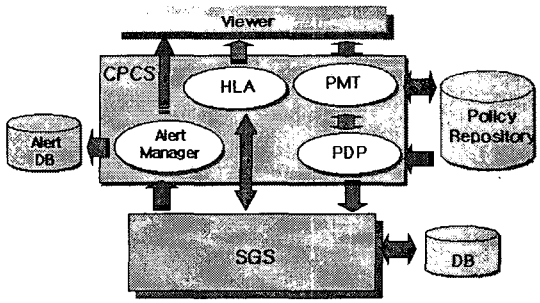
[그림 1] 정책기반 네트워크보안시스템의 구조

각 보안 게이트웨이 시스템은 외부 네트워크에서 내부 네트워크로의 패킷을 분석하고 공격을 탐지하여 사이버 순찰 제어 시스템으로 경보를 올려주며, 사이버 순찰 제어 시스템이 정책을 생성하는데 기반이 되는 트래픽 정보와 로그 정보를 알려준다. 사이버 순찰 제어 시스템은 여러 보안 게이트웨이 시스템이 전달하는 트래픽 정보, 로그 정보, alert 정보를 이용하여 각 보안 게이트웨이 시스템이 탐지하지 못하는 부분까지도 탐지하여 보안 게이트웨이 시스템에 정책을 지시하는 기능을 가진다.

보안 게이트웨이 시스템은 센서(Sensor), 분석기(Analyzer), 사이버 순찰 에이전트(Cyber Patrol Agent)의 세 모듈로 구성되어 있고, 사이버 순찰 제어 시스템은 정책 관리 툴(Policy Management Tool-PMT), 정책 결정 모듈(Policy Decision Point-PDP), 경보 관리기(Alert Manager-AM), 상위레벨 분석기(High Level Analyzer-HLA)의 네 모듈로 구성되어 있다. 먼저, 센서는 외부 네트워크에서 내부 네트워크로 들어오는 패킷들을 카피하고, 패킷 데이터에서 필요한 정보만을 추출하는 축약 처리를 한다. 분석기는 사이버 순찰 제어 시스템에서 전송되어 데이터베이스에 저장된 정책 데이터와 축약된 정보를 비교 분석하여 침입을 탐지한다. 사이버 순찰 에이전트는 분석기에서 침입을 탐지하였을 때 침입에 해당하는 정보들을 기반으로 사이버 순찰 제어 시스템에 전송할 경보 데이터를 구성하거나 센서에게 침입에 해당하는 세션이나 패킷을 드랍하는 등 탐지에 대한 대응 행동을 한다.

정책 관리 툴은 정책 저장고(Policy Repository-PR)를 초기화하고 변경하는 기능을 가지는데 이러한 기능을 수행할 때마다 정책 결정 모듈에 정책 저장고가 변경되었음을 알려준다. 또한 정책 결정 모듈은 정책 수행 시 문제점이 발생되었을 때 뷰어(Viewer)에 전달하는 기능도 수행한다. 정책 결정 모듈은 정책을 결정하는 역할을 하며, 정책 저장고에 저장된 정책 데이터들을 해당 보안 게이트웨이 시스템에 전달하는 정책을 수행한다. 경보 관리기는 여러 보안 게이트웨이 시스템에서 올라온 경보 데이터를 경보 데이터베이스에 저장하고, 경보 데이터를 분석하여 관리자 뷰어에 경보 분석 정보를 전송한다. 사이버 순찰 제어 시스템의 상위레벨 분석기는 하나의 보안 게이트웨이 시스템에서 탐지하지 못하는 한계점을 극복하기 위한 것으로써 여러 보안 게이트웨이 시스템의 트래픽 정보와 로그 정보를 이용하여 분산된 공격을 탐지하는 역할을 수행한다.

구체적인 시스템의 구조는 다음 [그림 2]와 같다.



[그림 2] 시스템의 내부 구조

이와 같이 정책 기반의 네트워크 보안 시스템에서 모듈들 간의 통신 데이터들은 정책이다. 시스템 전반적인 모듈들 간에는 정책을 사용하기 때문에 PCIM을 바탕으로 한 확장된 네트워크 보안 정책 모델링의 구축이 필수적이다.

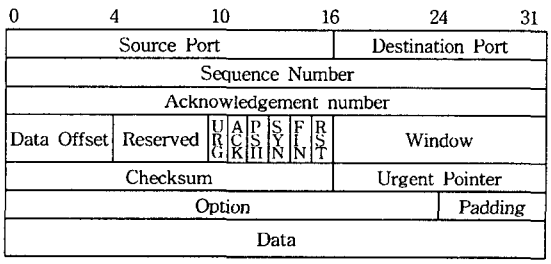
3. 네트워크 보안 정책 정보 모델을 위한 패킷 헤더 필드의 변수 정의

정책은 대부분 PolicyRule이 주를 이루고 있는데, PolicyRule은 어떤 조건을 만족하였을 때 이러한 행동을 취하라는 PolicyCondition과 PolicyAction으로

구성되어 있고, PolicyCondition은 PolicyVariable과 PolicyValue, 그리고 둘의 연산으로 구성된다. 침입 탐지 및 대응을 위해서는 PolicyRule을 구축해야 하는데, 공격 패턴을 볼 때 패킷 헤더 필드의 값들을 이용하여 공격을 탐지할 수 있다. 그러므로 침입 탐지를 위해서는 패킷 헤더 필드들을 변수화하는 것은 꼭 필요하다.

3.1. 패킷 헤더 필드의 변수 정의

네트워크 보안 정책 모델링을 위해 패킷 헤더 필드들을 변수화할 때 IP, TCP, UDP, ICMP 프로토콜별 패킷 헤더 필드들을 변수화하였다. 여러 프로토콜 패킷 헤더 중 [그림 3]은 TCP 프로토콜의 패킷 헤더 필드이며, 필드에 대한 변수 정의는 [표 1]과 같다. 나머지 프로토콜 패킷 헤더들도 동일하게 변수들을 정의할 수 있다.



[그림 3] TCP 프로토콜의 패킷 헤더 필드

[표 1] TCP 패킷 헤더 필드의 변수 정의

The Class "PolicySourcePortVariable"	
NAME	PolicySourcePortVariable
DESCRIPTION	The Source Port field of TCP or UDP Packet Header ALLOWED VALUE TYPES : PolicyIntegerValue (0 .. 65535)
DERIVED FROM	PolicyImplicitVariable
ABSTRACT	FALSE
PROPERTY	none
	.
	.
	.
The Class "PolicyTCPSynVariable"	
NAME	PolicyTCPSynVariable
DESCRIPTION	The SYN field of TCP Packet Header ALLOWED VALUE TYPES : PolicyBooleanValue
DERIVED FROM	PolicyImplicitVariable
ABSTRACT	FALSE
The Class "PolicyTCPFinVariable"	
NAME	PolicyTCPFinVariable
DESCRIPTION	The FIN field of TCP Packet Header ALLOWED VALUE TYPES : PolicyBooleanValue
DERIVED FROM	PolicyImplicitVariable
ABSTRACT	FALSE

3.2. 네트워크 보안 정책 정보 모델

본 장에서는 패킷 헤더 필드의 변수 정의를 이용하여 네트워크 보안 정책을 모델링한다. 네트워크 보안 정책은 외부 공격자의 공격을 탐지하였을 때 공격에 대한 대응방법으로 경보를 알리거나 패킷이나 세션을 드랍하는 정책, 메시지를 메일로 알리는 정책, 침입으로 판단되는 패킷의 속한 세션 정보를 로그로 저장하는 정책, 내부 네트워크 장치에 대응 행동을 내려주는 정책 등이 있다.

외부 공격에 대한 보안 정책의 한 예로 WhackAMole attack에 대한 정책 모델을 소개한다. 먼저 WhackAMole PolicyRule에 대한 간단한 설명을 하고, WhackAMole PolicyRule을 구성하고 있는 클래스들의 계층 구조를 소개한 후, WhackAMole의 UML Diagram을 소개한다.

3.2.1. WhackAMole PolicyRule

DefcoNet Pattern Signature Version 0.5 문서에는 침입 패턴을 정형화하고, 그에 대한 대응 방법을 명시하였다. 이 문서는 침입 탐지 및 대응 정책들을 모델링하기 위해 참조한다. 한 예를 보면 다음과 같은 침입 및 대응 정책이 존재한다.

[표 2] WhackAMole Pattern Signature

```
[ WhackAMole:TCPPort1;1;7;EMAIL|WPOP ]
  tcp any any > _homenet 12361
 (MESSAGE:Whack-a-mole Backdoor에 접근이 시도
  됨; FLAGS:S)
```

[표 2]에서 '[]'로 감싸인 부분은 패턴의 헤더 부분으로 패턴명, 분류명, 우선 순위, 침입 영향, 대응 방법의 순으로 되어 있다. 이 규칙은 프로토콜이 TCP이고, 목적지 IP 주소가 _homenet이고 목적지 포트 번호가 12361이며, 패킷 헤더의 Syn 플래그 필드가 TRUE일 때, Whack-a-mole Backdoor에 접근이 시도됨이라는 메시지를 전자우편로 보내거나 윈도우 화면에 팝업시키라는 것이다. 이 때 프로토콜이 TCP이고, 목적지 IP 주소가 _homenet이고 목적지 포트 번호가 12361이며, 패킷 헤더의 Syn 플래그 필드가 TRUE인 것은 PolicyCondition으로, 메시지를 전자우편이나 윈도우 화면에 팝업시키는 것은 PolicyAction으로 모델링할 수 있다.

3.2.2. WhackAMole PolicyRule의 클래스 계층 구조

WhackAMole PolicyRule의 모델링을 위한 클래스와 연관 클래스의 계층구조는 [표 3], [표 4]와 같다. [PCIMe]로 표시한 클래스들은 IETF의 정책 프레임워크 워킹 그룹이 제안한 PCIM의 확장 문서에 존재하는 클래스들이고, PCIM의 확장 문서에 존재하지 않지만 네트워크 보안 정책 정보 모델에 필요한 클래스들은 따로 정의하였고 [NSPIM]이라 표시하였다.

WhackAMole PolicyRule을 구성하고 있는 PolicyVariable은 [표 3]과 같이 4개이다. 4개의 PolicyVariable 중 PolicyTCPSynVariable은 TCP 패킷 헤더 필드를 변수화한 클래스이고, 나머지는 IP 패킷 헤더 필드를 변수화하였는데, [PCIMe]에 이미 정의되었기 때문에 [PCIMe]의 PolicyVariable을 사용하였다.

[표 3] WhackAMole의 클래스 상속 계층 구조

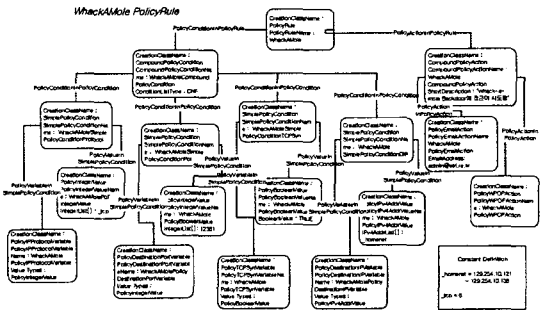
```
ManagedElement (abstract) [PCIMe]
    +---Policy (abstract) [PCIMe]
    |   +---PolicySet (abstract) [PCIMe]
    |   |   +---PolicyRule [PCIMe]
    |   +---PolicyCondition (abstract) [PCIMe]
    |   |   +---SimplePolicyCondition [PCIMe]
    |   |   +---CompoundPolicyCondition [PCIMe]
    |   +---PolicyAction (abstract) [PCIMe]
    |   |   +---SimplePolicyAction [PCIMe]
    |   |   +---CompoundPolicyAction [PCIMe]
    |   +---PolicyVariable (abstract) [PCIMe]
    |   |   +---PolicyImplicitVariable (abstract) [PCIMe]
    |   |   |   +---PolicyIPProtocolVariable [PCIMe]
    |   |   |   +---PolicyDestinationPortVariable [PCIMe]
    |   |   |   +---PolicyDestinationIPVariable [PCIMe]
    |   |   |   +---PolicyTCPSynVariable [NSPIM]
    |   +---PolicyValue (abstract) [PCIMe]
    |   |   +---PolicyIPv4AddrValue [PCIMe]
    |   |   +---PolicyIntegerValue [PCIMe]
```

[표 4] WhackAMole의 연관 클래스 상속 계층 구조

```
+---PolicyComponent (abstract) [PCIMe]
    |   +---PolicyConditionStructure (abstract) [PCIMe]
    |   |   +---PolicyConditionInPolicyRule [PCIMe]
    |   |   +---PolicyConditionInPolicyCondition [PCIMe]
    |   +---PolicyActionStructure (abstract) [PCIMe]
    |   |   +---PolicyActionInPolicyRule [PCIMe]
    |   |   +---PolicyActionInPolicyAction [PCIMe]
    |   +---PolicyVariableInSimplePolicyCondition [PCIMe]
    |   +---PolicyValueInSimplePolicyCondition [PCIMe]
    |   +---PolicyVariableInSimplePolicyAction [PCIMe]
    |   +---PolicyValueInSimplePolicyAction [PCIMe]
```

3.2.3. WhackAMole의 UML Diagram

WhackAMole PolicyRule의 인스턴스는 [그림 4]와 같이 UML Diagram으로 나타낼 수 있다. PolicyRule은 PolicyCondition과 PolicyAction으로 구성되어 있는데, WhackAMole PolicyRule은 프로토콜과 목적지 주소, 목적지 포트번호, TCPSyn 변수 값을 비교하는 복합적인 조건이고 전자우편을 보내고 윈도우 화면에 창을 팝업시키는 복합적인 행동이기 때문에 세 개의 SimplePolicyCondition 클래스와 연관되는 CompoundPolicyCondition 클래스와 두 개의 SimplePolicyAction 클래스와 연관되는 CompoundPolicyAction 클래스를 사용한다.



[그림 4] WhackAMole의 UML Diagram

4. 결 론

본 논문에서는 정책 기반의 네트워크 보안 시스템의 네트워크 보안 정책 정보 모델링을 위해 패킷 헤더 필드의 변수를 정의하였다. 내부 네트워크로 진입하는 패킷들에 공격 패턴이 있는지 탐지하기 위해서는 패킷 헤더의 필드들을 분석해야 하기 때문에 정책 모델을 구축하기 위해 패킷 헤더 필드들을 변수로 정의하는 것은 필수적이다. 이러한 패킷 헤더 필드의 변수는 공격 패턴을 탐지하고 대응하는 정책을 모델링하는데 유용하게 사용될 것이다.

본 논문에서 제안한 모델은 침입 탐지 및 대응을 위한 모델만을 제안하였기에 네트워크 자원을 위한 모델링 등 전체 시스템을 모델링하는데 많은 부분이 미약하다. 그러므로 계속적인 정책 기반의 네트워크 보안 시스템을 위한 모델이 연구되어야 하겠다.

참고문헌

[KYKIM] Ki Young Kim, et al., "A Policy-based Integrated Secure Architecture for Providing Security Service in WAN", The Sixth Conference on Communication Software, July 2001, pp35-39.

[SYKIM] Sook-Yeon Kim, et al., "Policy Core Information Model in Policy Based Network Management for Network Security", submitted.

[RFC3060] Strassner, J., and E. Ellesson, B. Moore, A. Westerinen, "Policy Core Information Model -- Version 1 Specification", RFC 3060, February 2001.

[PCIMe] B. Moore, and L. Rafalow, Y. Ramberg, Y. Snir, A. Westerinen, R. Chdha, M. Brunner, R. Cohen, J. Strassner, "Policy Core Information Model Extensions", <draft-ietf-policy-pcim-ext-02>, July 2001.