분산 침입탐지 노드간 통신 보안을 위한 키 관리 방안

박 눌[0] 정유석 홍만표
아주대학교 정보통신전문대학원
(nuri282[0], j8508, mphong)@ajou.ac.kr

Key Management Mechanism for Secure Communication of Distrubuted Intrusion Detecting Node

Nool Park[0] Yoo-Suk Jung Man-Pyo Hong
Graduate School of Information and Communication, Ajou University

요 약

기존 단일 침입탐지시스템의 단점을 해결하기 위한 노력으로 분산 침입탐지시스템에 관한 많은 연구가 진행중이다. 분산 침입탐지시스템은 다수의 침입탐지 에이전트들의 협력을 통해 침입을 판정하는 시스템으로, 에이전트간에 데이터의 전송이 요구된다. 더욱이 이들 데이터는 침입을 판정하기 위한 중요한 자료로, 데이터의 보안은 필수적이다. 하지만 지금까지의 대부분의 시스템들에서는 이 부분에 대한 해결책이 아직 미비한 상태이다. 본 연구에서는 분산 침입탐지 시스템의 성능 저하를 최소화하면서 전송 데이터를 보안하기 위한 새로운 키 관리 시스템을 제안하고자 한다.

1. 서론

정보통신의 비약적인 발전에 따른 정보화가 진행되면서 정보화의 역기능을 막기 위한 정보보호의 문제가 심각하게 대두되었다. 이에 따라 인터넷 망의 무수한 침입자들로부터 시스템을 보호하기 위한 침입탐지시스템에 관한 연구가 활발히 진행되었다.

초기의 침입탐지시스템은 보호 대상 호스트 및 네트워크에서 데이터를 수집하여 침입을 판정하기까지의 일련의 과정이 단일 시스템에서 이루어지는 중앙 집중 시스템이었다. 하지만 최근 네트워크 공격의 변화에 따라 침입탐지시스템의 형태도 변화되고 있다. AAFID[1], GrIDS[2]으로 대표되는 분산 침입탐지시스템이 그것으로 침입을 탐지하기 위한 다수의 컴포넌트들로 하나의 시스템이 구성된다[1]. 각각의 컴포넌트(이후 에이전트)들은 자신이 설치된 호스트 혹은 네트워크로부터 데이터를 수집 및 분석하고 그 결과를 상위 에이전트[1, 2]나 동일한 역할을 하는 다른 에이전트[3]에게 전송하여, 그 합쳐진 결과로부터 최종 침입을 판정하게 된다. 따라서 분산 침입탐지시스템은 중앙 집중식 침입탐지시스템에서와는 달리 에이전트들간의 데이터 전송이 요구된다. 더욱이 이 때 전송되는 데이터는 침입을 판단하는 데 사용되는 중요한 자료이기 때문에 침입탐지시스템의 오동작을 막기 위한 데이터의 보안은 필수적이라고 할 수 있다. 하지만 지금까지의 대부분의 연구는 분산 침입탐지시스템의 전체적인 구조에 초점이 맞추어진 채, 시스템을 구성하는 에이전트간의 통신 보안에 대해서는 거의 다루어지지 않았다. 따라서 본 논문에서는 분산 침입탐지시스템에서의 데이터 보안을 위하여 요구되는 사항들을 고려하여 새롭게 제안된 키 관리 구조에 대해 살펴보도록 하겠다.

본 논문은 한국전자통신연구원의 지원에 의한 것임

2. 관련 연구

2.1 키 관리 요구 사항

네트워크의 다수의 호스트에 분포되어 있는 분산 침입탐지 에이전트들간에 침입에 관한 정보를 공유하기 위해서는 통신은 필수적이다. 하지만 침입탐지 에이전트를 직접 공격하는 것보다 에이전트간 통신을 공격하는 것이 훨씬 쉽기 때문에 그만큼 공격자의 공격 대상이 되기 쉽다[8]. 따라서 이를 보호하기 위해서 전송되는 데이터의 암호화가 요구되며, 이 때 사용되는 키는 매우 중요한 요소라고 할 수 있다. 다음은 이를 위해 요구되는 사항을 기술한 것이다.

- 인증 : 통신에 참여하기 위해서는 제 3자에 의해 생성된 에이전트가 아닌 합법적인 침입 탐지 에이전트임을 확인할 수 있는 상호 인증 과정이 선행되어야 한다.
- 비밀성 : 에이전트들간의 메시지는 불법적인 제 3자로부터 보호되어야 한다.
- 무결성 : 에이전트간에 전송되는 메시지는 전송도중에 불법적인 제 3자에 의해서 위조 및 변경되어서는 안 된다.

지금까지 분산 침입탐지 시스템 내 통신 보안을 위한 키의 요구사항에 대해 알아보았다. 하지만 한 가지

더 고려해야 할 사항은 키를 사용한 암호화 통신이 역으로 침입탐지시스템의 성능을 저하시키는 원인이 되서는 안된다는 점이다. 따라서 암호화 통신을 하면서 침입탐지 시스템의 성능 저하를 최소화시킬 수 있는 방법이 요구된다.

2.2 그룹키(Group Key)

본 시스템에서는 에이전트들의 부담을 최소화하기 위한 방법으로 그룹키를 이용한다. 그룹키는 원격회의나 소프트웨어 배포 등 멀티캐스트 보안을 위해 사용되는 그룹 내 세션키로 통신에 참여하는 모든 그룹 멤버에게 배포된다. 일대일 통신과 달리 그룹 멤버가 변하더라도 세션이 종료되는 것이 아니기 때문에, 멤버쉽의 변화에 따라 키가 갱신되어 현재의 그룹 멤버에게 전송된다.

2.2 기존의 그룹키 관리 방식

선형 네트워크 구조에 적용 가능한 Clique[5]은 Diffie-Hellman 방식을 그룹에 적용한 GDH[5] 방식을 적용한 방법으로 모든 그룹 멤버가 키 생성에 참여한다. 따라서 키를 생성할 때마다 전 멤버가 키 생성 과정에 참여해야 하는 번거로움이 발생한다[4]. 반면 Iolus[6]는 하나의 그룹을 여러 개의 서브그룹으로 나누어 서브그룹별로 키를 관리하기 때문에 Clique의 단점을 일부 해결한다. 하지만 전체 그룹내에 메시지를 전송하기 위해서는 서브 그룹간 관리자에 의해 그룹 수만큼 메시지의 암/복호화가 추가로 요구된다. 또한 그룹을 도메인 형식으로 구성하는 GKMP[7] 방식 역시 Iolus와 같이 매번 암/복호화 과정을 추가로 수행해야 하는 단점이 존재한다. 따라서 이러한 문제점들을 해결하면서 키를 유지하는 비용을 최소화 할 수 있는 새로운 방법이 요구된다.

3. 제안 시스템

본 시스템에서는 그룹키를 이용하면서 에이전트 시스템에 주는 로드를 최소화 할 수 있는 새로운 그룹 키 관리 방식을 제안한다.

3.1 시스템 계수

- Agent : 분산 침입탐지시스템의 컴포넌트로 침입을 판정하기 위한 데이터를 수집 및 분석한다. 분석된 결과를 분배된 키를 통해 암호화하여 다른 에이전트에게 전송한다.
- CA(Certificate Authority) : Agent에게 인증서를 생성/발급하고 이를 관리한다.
- GKMA(Group Key Management Agent) : 그룹키를 생성하여 Agent에게 분배하고 이를 관리한다.
- GK(Group key) : 에이전트들이 공유하는 세션키로, 다음과 같은 메시지 형식으로 전송된다.

ID	Version	Type	Key

  - ID : key 식별자
  - Version : key 갱신 정보를 구별하기 위한 식별자
  - Type value : 1 = initial key
    2 = join
    3 = leave
    4 = key renewal
  - Key : 그룹키
- A_{pub}, A_{priv} : Agent의 public key, private key
- KMA_{pub}, KMA_{priv} : KMA의 public key, private key

3.2 키 관리 프로토콜

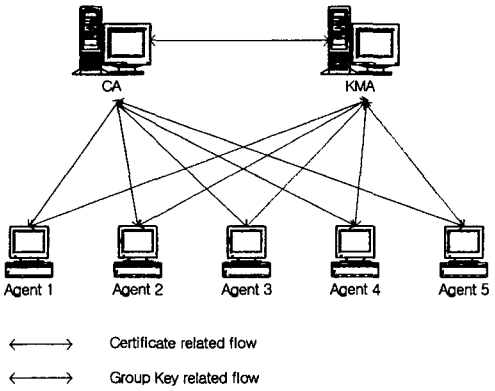

그림 1. 키 관리 시스템 구조

3.2.1 인증서 발급 단계

KMA와 에이전트는 인증서 요청/신분 확인/키 쌍 생성 등의 과정을 통하여 CA로부터 인증서를 발급받는다.

3.2.2 그룹 초기화 단계

① 인증서를 발급받은 에이전트들은 자신의 인증서와 id를 가지고 KMA에게 그룹 키 발급을 요청한다.

$$E_{Apriv}(id) + Certificate$$

② KMA는 agent의 인증서를 확인하고 그룹 키를 생성한다.

③ 그룹 키를 분배한다.

$$E_{Apub}(GK)$$

3.3.3 새로운 에이전트의 추가 단계

새로운 에이전트가 추가될 경우 새로운 그룹 키를 생성하여 추가된 에이전트 및 기존 에이전트들에게 전송한다.

① KMA는 새로운 에이전트가 추가되었음을 기존 에이전트들에게 알린다. 이 때 갱신한 키는 전송되지 않고, 나머지 세 필드만으로 키가 갱신되었음을 알린다.

ID	Version + 1	2

② KMA로부터 메시지를 받은 기존의 에이전트들은 Hash 함수를 통해 새로운 키를 생성한다.
$GK_{new} = Hash(GK)$
$Version = Version + 1$

③ KMA는 새롭게 추가된 에이전트에게 그룹 키를 분배한다.

3.3.4 기존 에이전트 제거 단계

KMA는 에이전트가 제거되었음을 알리고 그룹 초기화 단계를 반복한다.

에이전트가 공격자에 의해 수정된 경우, 에이전트를 그룹에서 제거시킴과 동시에 CA를 통해 인증서를 폐지시킨다.

3.3.4 키 갱신

멤버쉽의 변화없이 주기적이 키 갱신이 필요한 경우 Type 필드의 값을 4로 하여 새로운 에이전트 추가 단계를 반복한다.

3.3.5 전송 단계

메시지를 전송하고자 하는 에이전트는 메시지를 그룹키로 암호화하여 전송한다. 이 때 ID, Version 정보를 함께 실어 보내어 key의 동기를 맞춘다.

4. 제안 시스템 분석

4.1 특징

본 제안 시스템은 다음과 같은 특징을 갖는다.

① PKI를 적용하여 모든 에이전트가 CA를 통한 인증 절차를 거치므로, 인증서를 갖는 에이전트를 신뢰할 수 있다. 또한 각 에이전트의 비밀키로 암호화하여 그룹키를 분배하므로, 해당 에이전트만이 키를 복호화할 수 있다.

② 메시지를 암/복호화하는 데 그룹 키를 이용하므로 에이전트가 증가하더라도 사용되는 키의 수가 늘어나지 않는다. 또한 단 한번의 암/복호화만으로 메시지의 통신이 가능하다.

③ 메시지를 전송할 때 사용되는 키의 ID와 Version 정보를 실어 보내어, 그룹 멤버간의 키의 동기를 맞출 수 있다.

4.2 기존의 그룹 키 관리 방식과의 비교

기존의 그룹 키 관리 방식에서는 새로운 에이전트가 추가되는 경우 새로운 키를 생성하여 모든 그룹 멤버에게 분배한다[5, 6, 7]. 하지만 제안된 시스템에서는 기존의 키를 해쉬하여 사용하도록 하여 각 에이전트들에서 수행되는 계산량을 최소화하였다. 해쉬함수는 단방향 함수이므로, 새로 추가된 에이전트가 새로운 키를 받더라도 예전의 키를 알아낼 수 없을 뿐더러 기존의 에이전트들은 새로운 키를 전송받지 않더라도 스스로 키를 갱신할 수 있다. 따라서 키를 수신하기 위한 암/복화 과정을 줄일 수 있을 뿐 아니라, 키 자체가 전송되지 않기 때문에, 키가 공격자에게 노출될 위험이 줄어들게 된다.

5. 결론

분산 침입탐지 에이전트간 키를 이용한 암호화 통신을 할 경우, 공격자로부터 취약한 통신 상의 위협을 줄일 수 있다. 하지만 지금까지 분산 침입탐지시스템의 문제점으로 거론되었던 처리 속도의 지연을 고려해 볼 때 암호화 통신을 하되, 전체 시스템에 주는 부담을 최소화할 수 있는 방법이 요구된다.

본 논문에서는 분산 침입탐지 에이전트를 위한 그룹 키 관리 시스템에 대해 제안하였다. 그룹키를 이용한 암호화 방식은 메시지의 암/복호화에 드는 에이전트들의 계산 비용을 줄일 수 있으며, 해쉬 함수를 이용한 키의 갱신은 그룹키를 사용하는 데 있어 드는 부담을 최소화할 수 있다. 하지만 이 시스템에서는 모든 에이전트들을 하나의 그룹으로 관리하므로 확장성의 문제가 발생할 수 있다. 따라서 앞으로 이를 해결할 수 있는 방안을 생각해 볼 수 있을 것이다.

6. 참고문헌

[1] E. Spafford and D. Zamboni, "Intrusion Detection using Autonomous Agents", Computer Networks 34, pp. 547-570, 2000.

[2] S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, J. Rowe, S. Staniford-Chen, R. Yip, and D. Zerkle. "The Design of GrIDS: A Graph-Based Intrusion Detection System." Technical Report CSE-99-2, U.C. Davis Computer Science Department January 1999

[3] Gregory B. White, Eric A. Fisch and Udo W. Pooch, "Cooperating Security Managers: A Peer-Based Intrusion Detection System", IEEE Network, January/February 1996, pp.20-23

[4] 박희운, 이염영, 박원주, 이종태, 손승원, "확장성을 제공하는 안전한 멀티캐스트 키 관리 구조", 정보과학회논문지: 정보통신 제 29 권 제 2 호(2002. 4)

[5] M. Steiner, G. Tsudik, and M. Waidner, "Key agreement in dynamic peer groups", IEEE Transactions on Parallel and Distributed Systems, August 2000

[6] H. Harney and C. Muckenhirn, "Group Management Protocol(GKMP) Architecture, IETF RFC 2094, 1997

[7] S. Mittra, "Iolus : A Framework for Scalable Secure Multicasting", 1997

[8] Rajeev Gopalakrishna, "A Framework for Distributed Intrusion Detection using Interest-Driven Cooperative Agents", CERIAS Tech Report 2001-44