아이덴티티를 이용한 능동 노드들간의 접근 정책 전송 방법

김영수*, 한종옥*, 서동일*, 손승원*
*한국전자통신연구원, 네트워크보안연구부

Transferring Access Policies Between Active Nodes Using Identities

Youngsoo Kim,Jongwook Han*, Dongil Seo*, Seung-won Son**
**Network Security Department, ETRI*

요 약

액티브 네트워크 환경에서 액티브 노드(라우터 또는 스위치)의 기능은 액티브 익스텐션(active extension)에 의해 다이나믹하게 확장될 수 있다. 즉, 액티브 노드 기능을 확장하는 소프트웨어 모듈을 " 익스텐션" 이라 할 수 있다. 이러한 융통성(flexible)이 강한 구조는 새로운 네트워크 프로토콜과 서비스들을 활성화시키고 있으나, 다른 한편으로는 매우 심각한 안전성(safety)과 보안(security) 문제를 야기시키게 된다. 본 논문에서 다루는 액티브 익스텐션 관련 보안 문제는 하나의 액티브 노드 상의 익스텐션이 다른 액티브 노드에 접근하려 할 경우 이에 대한 제어 방법이 반드시 제공되어야 한다는 것이다. 특히, 이 문제에서는 액티브 노드들간의 인증(authentication)이 매우 중요하다. 여기에서는 액티브 노드들간의 접근 정책 전송을 위해 각 객체가 갖는 고유 정보인 아이덴티티(identity)를 이용한다. 우리는 본 논문에서 아이덴티티를 통하여 액티브 노드들간의 인증(authentication)을 행하는 새로운 방식의 접근 정책(access policy) 전송 방법을 제안한다.

I. 서론

액티브 네트워크 환경에서 액티브 노드의 기능은 액티브 익스텐션(active extension)에 의해 다이나믹하게 확장될 수 있다. 즉, 액티브 노드 기능을 확장하는 소프트웨어 모듈을 "익스텐션" 이라 할 수 있다. 이러한 융통성이 강한 구조는 새로운 네트워크 프로토콜과 서비스들을 활성화시키고 있으나, 다른 한편으로는, 매우 심각한 안전성(safety)과 보안(security) 문제를 야기할 수 있으므로, 이에 대하여 매우 신중하게 고려되어야 한다[1][2]. 액티브 익스텐션과 관련하여 고려되어야 할 보안 문제들 중 하나는 어떠한 방법으로 액티브 익스텐션이 액티브 노드 상에서 접근할 수 있는 리소스들과 데이터들을 제어할 것인가이다. 이 문제와 관련하여서는, 접근 제어 정책(access control policy)을 이용하는 방법들이 제안되고 있다. 한편, 하나의 액티브 노드 상의 익스텐션이 다른 액티브 노드의 리소스에 접근하려 할 경우 이를 어떠한 방법으로 제어할 것인가도 심각하게 고려되어야 할 사항이다. 특히, 이 문제에서는 액티브 노드들간의 인증(authentication)이 매우 중요하다. 본 발명은 이 두번째 문제 해결에 관한 것으로, 기존의 Kerberos[3]등을 이용하는 방법과는 다른 각 객체의 아이덴티티(identity)를 이용하는 방법을

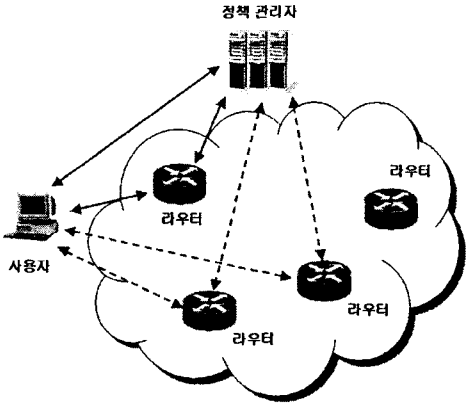
제안한 것이다. 제 2장에서는 액티브 노드와 관련한 보안 문제에 대하여 다루고, 제 3장에서는 이러한 문제를 해결하기 위해 일반적으로 사용되는 Kerberos를 이용한 접근 정책 전송 방식에 대하여 살펴본다. 제 4장에서는 기존의 방식을 개선한 새로운 방식의 접근 정책 전송 방식을 제안한다.

II. 액티브 노드의 보안 문제

본 논문에서는 액티브 노드가 갖는 보안(security) 문제를 액티브 노드 구조에 초점을 맞춘 내부적 관점과 액티브 노드들간의 통신에 초점을 맞춘 외부적 관점으로 나누어 고찰한다.

보안(security) 문제는 액티브 노드 익스텐션이 리소스 사용이나 데이터 읽기/쓰기를 통해 다른 사용자들이 받는 네트워크 서비스들을 방해하는 것을 막는 것이다. 이 문제는 기존의 운영 체제가 다루고 있는 문제점과 유사하나, 액티브 노드는 매우 다른 작업을 수행한다. 즉, 액티브 노드의 역할은 패킷을 포워딩하고 프로세싱하는 것이지, 범용의 데이터 프로세싱이나 저장소 관리, 또는 사용자 인터페이스 지원등이 아니라는 것이다. 이는 액티브 노드 운영 체제는 기존과 다른 보안 사항들을 고려해야 한다는 의미이다. 자신들만의 특수한 작업(QoS 구현, 액티브 노드 경로 선택, 또는 데이터 암호화 등) 수행을 위해 액티브 노드 익스텐션들은 링크 대역폭(link bandwidth) 같은 액티브 노드 리소스를 제어하거나 라우팅 테이블 같은 데이터 구조에 접근 한다. 익스텐션은 또한 패킷을 버리거나 패킷 내용을 변경하는 등의 데이터 트래픽 조작을 행한다. 적절한 보안 고려 사항이 없다면, 이러한 동작들이 다른 사용자들에게 해를 끼칠 수 밖에 없다는 것은 자명한 일이다. 위에서 언급한 액티브 노드 리소스(예를 들면, 링크 대역폭 또는 라우팅 테이블)나 액티브 노드들을 지나는 데이터 트래픽 등의 리소스들이 오용되고 악의적으로 사용되는 것을 막기 위해, 액티브 액티브 노드는 적절한 접근 제어 메커니즘을 가져야 한다.

액티브 익스텐션이 갖는 위협은 단일 액티브 노드의 로컬 리스크 뿐 아니라, 전체 네트워크 범위로 미칠 수 있다. 적절한 접근 제어 없이, 익스텐션은 대역폭 점유 매개변수를 증가시키거나, 또는 자신의 플로우들을 다른 사용자들의 리소스 액티브 노드와 결합시킴에 의해 대역폭을 가로챌 수 있다[5]. 또한, 악의적으로 만들어진 익스텐션들은 다른 사용자들의 트래픽을 방해하기 위해 랜덤 플로우들의 경로를 변경하거나, 특정 서버로 플로우들을 우회시킴으로써 서비스 거부 공격(Denial-of-Service attack)을 행할 수 있다. 링크 대역폭 관리와 트래픽 제어 외에도, 익스텐션은 표준 소켓-기반 통신 메커니즘들을 이용하여 애플리케이션이나 다른 익스텐션들과 통신이 가능하다.



[그림 1] 외부 보안 관점의 시스템 구조 예

III. 기존의 접근 정책 전송 방식

본 장에서는 액티브 노드들간의 통신 관점에서의 문제점 해결을 위한 방법으로 기존의 Kerberos를 이용한 접근 정책 전송 방식에 대하

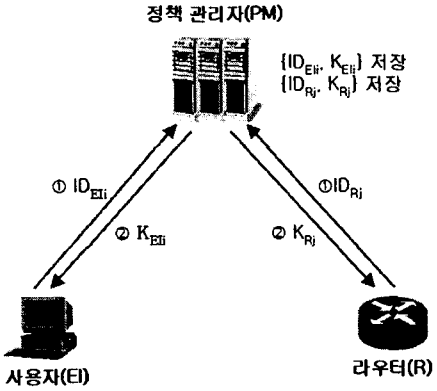
여 살펴본다.

[그림 1]에서 사용자(user)는 익스텐션 코드를 액티브 노드에 주입하는 객체로서, 익스텐션 초기화자(Extension Initiator, EI)로 볼 수 있다.

하나의 도메인(네트워크 또는 서브 네트워크) 상에는 하나의 정책 관리자(Policy Manager, PM)가 존재하고, 정책 관리자는 신뢰할 수 있는 기관(trusted authority)이라 가정한다. 정책 관리자는 이 도메인내에 존재하는 모든 액티브 노드와 그들의 연결 상태를 모두 알고 있고 제어가 가능하다. 네트워크 관리자는 정책 정의 언어를 이용하여 네트워크 상의 익스텐션들을 위한 정책을 정의할 책임을 진다. 도메인 상의 사용자들은 액티브 노드들에 익스텐션들을 인스톨하기 이전에 PM에 등록해야한다.

PM의 주된 역할은 정책에 대한 집행, 즉 네트워크 관리자가 정한 정책에 기반하여 익스텐션들이 갖는 접근 허가 권한을 부여하는 것이다. 정책 서버와 같은 역할 이외에 PM은 또한 다음 절에서 설명되는 보안 프로토콜상에서 인증 센터(authentication center)와 키 분배 센터(key distribution center)로서의 역할을 한다. 이 보안 프로토콜은, 메시지 무결성과 기밀성 보장을 통해, 정책과 익스텐션 코드들이 액티브 노드에 안전하게 전송되도록 하며, 또한 PM과 액티브 노드들로 하여금 사용자들을 인증할 수 있도록 한다.

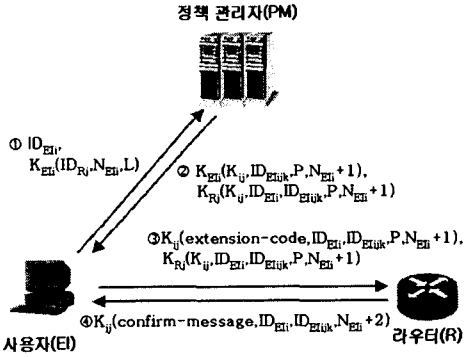
본 예에서의 안전한 접근 정책 전송은 Kerberos에 기반한다. Kerberos는 사용자들이 PM을 통하여 액티브 노드들과 인증을 행하도록 하고, 익스텐션 코드와 해당 접근 정책 정보들이 액티브 노드에 안전하게 전송되도록 한다. 우선 하나의 도메인에는 n개의 사용자 또는 익스텐션 초기화자(Extension Initiator, EI)와 m개의 액티브 노드가 존재한다(EI_1, … , EI_n, R_1, … , R_m).



[그림 2] 사전 등록/키분배 과정의 예

[그림 2]는 PM과 EIi, PM과 Rj간의 사전 등록 및 키분배 과정을 나타낸 그림이며, [그림 3]은 EIi가 익스텐션 코드(extension codes)를 액티브 노드(R_j)에 인스톨하는 접근 정책 전송 예를 나타낸다.

사전 등록 및 키분배 과정은 다음과 같다



[그림 3] 접근 정책 전송의 예

• EI -> PM: {ID_{EIi}} / R -> PM: {ID_{Rj}}

익스텐션 초기화자 EIi는 자신이 속한 도메인의 정책 관리자(PM)에게 자신의 아이덴티티를 등록하고 비밀키를 발급받기 위해 {ID_{EIi}}를 PM에게 전송한다. 이와 마찬가지로, 액티브 노드 Rj도 자신이 속한 도메인의 정책 관리자(PM)에게 자신의 아이덴티티를 등록하고 비밀키를 발급받기 위해 {ID_{Rj}}를 PM에게 전송한다.

• PM -> EI: {K_{EIi}} / PM -> R: {K_{Rj}}

EIi와 Rj의 아이덴티티를 받은 PM은 각각의 비밀키들을 생성하여 자신의 데이터베이스에 {ID_{EIi}, K_{EIi}}와 {ID_{Rj}, K_{Rj}} 쌍을 저장하고, EIi와 Rj에게 각각의 비밀키를 전송한다.

접근 정책 전송 과정은 다음과 같다

• EI -> PM: {ID_{EIi}, K_{EIi}(ID_{Rj}, N_{EIi}, L)}

익스텐션 초기화자는 PM에게 {ID_{EIi}, K_{EIi}(ID_{Rj}, N_{EIi}, L)}를 보낸다. 여기서 ID_{EIi}는 익스텐션 Eii아이덴티티이고, ID_{Rj}는 목적 액티브 노드 Rj의 아이덴티티이다. 그리고, N_{EIi}는 재전송 공격을 막기 위해 EIi가 선택한 랜덤 수이고, L은 인스톨된 익스텐션이 액티브 노드 Rj에 접근할 리소스들의 리스트이다. 이 메시지는 EIi가 PM에게 자신을 확인시키는데 사용된다. 이를 수신한 PM은 사전에 EIi와 나누어 가진 동일한 대칭키 방식 비밀키(K_{EIi})를 이용하여 이 메시지를 부분적으로 복호화한다. 이 메시지가 성공적으로 복호화되면, PM은 메시지 내용을 신뢰하게 되고, 난수 N_{EIi}를 통해 재전송 공격이 방지되었음을 확인하게 된다. 또한, 이 메시지는 액티브 노드 Rj와 통신하기 위한 세션키와 접근 정책을 요청하는데 사용된다. 이 메시지를 수신한 PM은 글로벌한 차원에서 유일한 아이덴티티를 생성한다. 즉, EIi가 Rj상에 k번째로 생성한 익스텐션을 의미하는 ID_{EIijk}를 생성한다. 그리고, EIi와 Rj간의 통신을 위한 세션키 K_{ij}와 해당 익스텐션을 위한 접근 정책 P, 리소스 리스트 L 등을 생성한다. PM은 다음과 같은 회신 메시지를 보낸다.

• PM -> EI: {K_{EIi}(K_{ij}, ID_{EIijk}, P, N_{EIi} + 1)}, {K_{Rj}(K_{ij}, ID_{EIi}, ID_{EIijk}, P, N_{EIi} + 1)}

이 메시지는 두 부분으로 나누어지는데, 첫번째 부분은 PM이 갖고 있는 EIi의 비밀키(K_{EIi})로 암호화된다. EIi는 세션키, 익스텐션 식별 정보, 접근 정책, 그리고 난수 등을 얻기 위해 이를 복호화한다. 두번째 부분은 세션키와 익스텐션 식별 정보, 정책 등을 담고 있는 부분으로 PM이 갖고 있는 액티브 노드 Rj의 비밀키(K_{Rj})로 암호화된다.

• EI -> R: {K_{ij}(extension-code, ID_{EIi}, ID_{EIijk}, P, N_{EIi}+1)}, {K_{Rj}(K_{ij}, ID_{EIi}, ID_{EIijk}, P, N_{EIi}+1)}

이 메시지도 두 부분으로 구분된다. 첫번째 부분은 EIi의 식별 정보, 익스텐션 코드, 난수 등으로 이루어지고 세션키(K_{ij})로 암호화된다. 성능 향상 차원에서 해쉬 함수를 이용하여, 익스텐션 코드를 메시지 다이제스트(message digest) 형태로 변경하거나, (기밀 사항을 담고 있지 않았다면) 평문 형태로 암호화 없이 전송될 수도 있다. 또한, 익스텐션 코드를 직접 보내는 대신, 이것이 포함된 안전한 코드 서버를 참조할 수 있도록 하는 것도 하나의 방법이 될 수 있다. 두번째 부분은 PM으로부터 받은 내용과 동일하다.

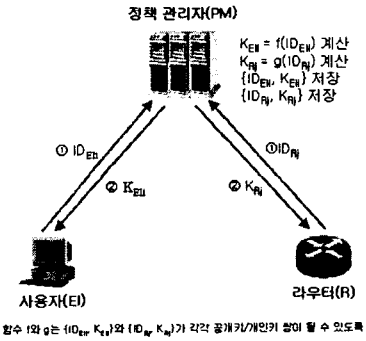
• R -> EI: {K_{ij}(confirm-message, ID_{EIi}, ID_{EIijk}, N_{EIi} + 2)}

액티브 노드 Rj는 익스텐션 인스톨의 성공 여부를 EIi에게 알려주기 위해 이 메시지를 전송한다

IV. 제안하는 방식

여기서 제안하는 방식은 액티브 노드들간의 통신 관점에서의 보안 문제 해결 방법에서, 접근 정책 전송 방식을 개선한 것으로, 대칭키 암호 방식이 아닌 공개키 암호 방식을 사용하면서도, 기존의 공개키 암호 방식의 문제점 해결을 위해 아이덴티티를 이용한 새로운 방식을 제시하고자 한다[6][7]. 제안하는 방식에서의 정책 관리자는 정책 서버와 같은 역할 이외에 PM은 또한 다음 절에서 설명되는 보안 프로토콜상에서 키 분

배 센터와 키 생성 센터로서의 역할을 한다. 이 보안 프로토콜은, 메시지 무결성과 기밀성 보장을 통해, 정책과 익스텐션 코드들이 액티브 노드에 안전하게 전송되도록 하며, 또한 PM과 액티브 노드들로 하여금 사용자들을 인증할 수 있도록 한다. [그림 4]는 PM과 EIi, PM과 Rj간의 사전 등록 및 키분배 과정을 나타낸 그림이며, [그림 5]은 EIi가 익스텐션 코드(extension codes)를 액티브 노드(R_j)에 인스톨하는 접근 정책 전송 과정을 나타낸다.



단, 여기서 함수 f와 g는 {ID_{EIi}, K_{EIi}}와 {ID_{Rj}, K_{Rj}}가 각각 공개키/개인키 쌍이 될 수 있도록 하는 함수

[그림 4] 제안 시스템의 사전 등록/키분배 과정

사전 등록 및 키분배 과정은 다음과 같다.

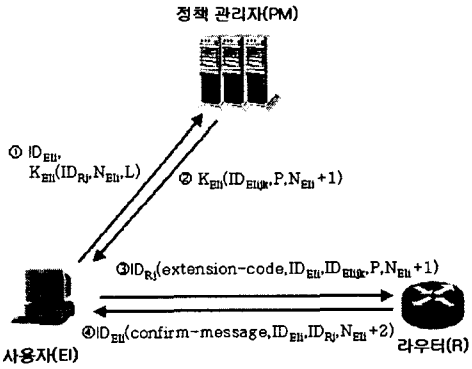
• EI -> PM: {ID_{EIi}} / R -> PM: {ID_{Rj}}

익스텐션 초기화자 EIi는 자신이 속한 도메인의 PM에게 자신의 아이덴티티이자 공개키로 사용될 {ID_{EIi}}를 전송하여 등록한다. 이와 마찬가지로, 액티브 노드 Rj도 자신이 속한 도메인의 PM에게 자신의 아이덴티티이자 공개키로 사용될 {ID_{Rj}}를 전송하여 등록한다.

• PM -> EI: {K_{EIi}} / PM -> R: {K_{Rj}}

EIi와 Rj의 아이덴티티를 받은 PM은 다음과 같이 각각의 비밀키들을 생성하여 자신의 데이터베이스에 {ID_{EIi}, K_{EIi}}와 {ID_{Rj}, K_{Rj}} 쌍을 저장하고, EIi와 Rj에게 각각의 비밀키를 전송한다. (단, 여기서 함수 f와 g는 {ID_{EIi}, K_{EIi}}와 {ID_{Rj}, K_{Rj}}가 각각 공개키/개인키 쌍이 될 수 있도록 하는 함수이다)

K_{EIi} = f(ID_{EIi}), K_{Rj} = g(ID_{Rj})



[그림 5] 제안 시스템의 접근 정책 전송 과정

접근 정책 전송 과정은 다음과 같다.

• EI -> PM: {ID_{EIi}, K_{EIi}(ID_{Rj}, N_{EIi}, L)}

익스텐션 초기화자는 PM에게 {ID_{EIi}, K_{EIi}(ID_{Rj}, N_{EIi}, L)}를 보낸다. 여기서 ID_{EIi}는 공개키로 사용되는 익스텐션 EIi의 아이덴티티이고, ID_{Rj}는 목적 액티브 노드 Rj의 공개키로 사용되는 아이덴티티이다. 그리고, N_{EIi}는 재전송 공격을 막기 위해 EIi가 선택한 랜덤 수이고, L은 인스톨된 익스텐션이 액티브 노드 Rj에 접근할 리소스들의 리스트이다. 이 메시지는 EIi가 PM에게 자신을 확인시키는데 사용된다. 이를 수신한 PM은 사전에 생성하여 EIi에게 발급한 EIi의 개인키(K_{EIi})를 이용하여 이 메시지를 부분적으로 복호화한다. 이 메시지가 성공적으로 복호화되면, PM은 메시지 내용을 신뢰하게 되고, 난수 N_{EIi}를 통해 재전송 공격이 방지되었음을 확인하게 된다. 또한, 이 메시지는 익스텐션 초기화자 EIi와 액티브 노드 Rj 사이의 접근 정책을 요청하는데 사용된다. 이 메시지를 수신한 PM은 글로벌한 차원에서 유일한 식별 정보를 생성한다. 즉, EIi가 Rj상에 k번째로 생성한 익스텐션을 의미하는 ID_{EIijk}를 생성한다. 그리고, 해당 익스텐션을 위한 접근 정책 P, 리소스 리스트 L 등을 생성한다. PM은 다음과 같은 회신 메시지를

보낸다.

• PM -> EI: {K_{EIi}(ID_{EIijk}, P, N_{EIi} + 1)}

이 메시지는 PM이 갖고 있는 EIi의 개인키(K_{EIi})로 암호화된다. EIi는 익스텐션 식별 정보, 접근 정책, 그리고 난수 등을 얻기 위해 이를 복호화한다.

• EI -> R: {ID_{Rj}(extension-code, ID_{EIi}, ID_{EIijk}, P, N_{EIi} + 1)}

이 메시지는 EIi의 식별 정보, 익스텐션 코드, 크레덴셜, 난수 등으로 이루어지고 액티브 노드 Rj의 공개키인 ID_{Rj}로 암호화된다. 성능 향상 차원에서 해쉬 함수를 이용하여, 익스텐션 코드를 메시지 다이제스트 형태로 변경하거나, (기밀 사항을 담고 있지 않았다면) 평문 형태로 암호화 없이 전송될 수도 있다. 또한, 익스텐션 코드를 직접 보내는 대신, 이것이 포함된 안전한 코드 서버를 참조할 수 있도록 하는 것도 하나의 방법이 될 수 있다.

• R -> EI: {ID_{EIi}(confirm-message, ID_{Rj}, ID_{EIijk}, N_{EIi} + 2)}

액티브 노드 Rj는 익스텐션 인스톨의 성공 여부를 EIi에게 알려주기 위해 EIi의 공개키 ID_{EIi}로 확인 메시지를 암호화하여 전송한다.

V. 결론

본 논문은 기존의 접근 정책 전송 방식을 개선한 것으로, 공개키 방식을 사용하면서도 공개키 디렉토리에 대한 별도 관리가 필요 없고, 프로토콜 자체가 매우 안전한 반면, 간단하고 경량화되어 실제 접근 정책 전송에 적용될 수 있을 것으로 기대된다

참고문헌

[1] K.Psounis, "Active networks: Applications, Security, Safety, and Architectures", IEEE Communications Surveys, 1999.

[2] Y.S.Kim, J.C.Na, S.W.Sohn, "A Secure Active Packet Transfer using Cryptographic Techniques", Journal of The Korean Institute of Information Security and Cryptology, pp.135-145, 2002.

[3] J.G.Steiner, B.C.Neuman, and J.I.Schiller, "Kerberos: An Authentication Service for Open Network Systems", Proceedings of Winter USENIX Conference, pp.191-201, 1988

[4] George Necula and Peter Lee, "Safe Kernel Extensions Without Run-Time Checking", In Proceedings 2nd Symposium on Operating Systems Design and Implementation (OSDI'96), pp.229-243, 1996

[5] I.Stoica, H.Zhang, and T.S.Eugen, "A Hierarchical Fair Service Curve Algorithm for Link-Sharing, Real-Time and Priority Service", Proceedings of the SIGCOMM'97 Symposium on Communications Architectures and Protocols, pp.249-262, 1998

[6] B.Schneier, "Applied Cryptography: Second Edition", Wiley, 1996.

[7] A.Shamir, "Identity-Based Cryptosystems and Signature Schemes", Proceedings of CRYPTO '84, Springer-Verlag, pp.47-53, 1985