액티브 네트워크에서의 협업적 구조를 통한 보안 강화

오하영° 채기준
이화여자대학교
hyoh°@ewhain.net, kjchae@ewha.ac.kr

Security Enforcement based on Cooperative Architecture in Active network

Hayoung Oh° Kijoon Chae
Ewha Womans University

요 약

기존의 패킷 교환 네트워크는 해킹과 같은 보안 공격에 많은 취약점을 가지고 있다. 침입차단 시스템 (Firewall system)과 침입탐지 시스템 (Intrusion Detection system)같은 보안 시스템이 개발되고 있지만 DOS나 Probe등을 비롯한 다양한 공격에 대해 적극적으로 대처 할 수 없다. 결과 DARPA를 비롯한 여러 기관에서 전송중인 액티브 패킷이 라우터에서 관리자의 정책을 담고 있는 코드를 실행할 수 있고 그 코드의 실행결과에 따라 라우터의 상태를 변경할 수 있는 액티브 네트워크 전반적인 구조를 제안하였다. 하지만 액티브 네트워크에서 중요한 것은 기존 네트워크와 달리 액티브 패킷이 액티브 노드의 자원에 접근함으로써 발생하게 되는 네트워크 보안이다. 따라서 액티브 노드의 NodeOs단에 Crypto engine, Integrity Engine, Authentication Engine, Authorization Engine등을 비롯한 액티브 노드 인증 및 액티브 패킷/코드 인증 보안 모듈을 둠으로써 액티브 노드 간 서로 안전한 협업적 관리를 통해 보안을 강화한다.

1. 서 론

기존의 패킷 교환 네트워크는 해킹과 같은 보안 공격에 많은 취약점을 가지고 있다. 침입차단 시스템 (Firewall system)과 침입탐지 시스템 (Intrusion Detection system)같은 보안 시스템이 개발되고 있지만 DOS나 Probe등을 비롯한 다양한 공격에 대해 적극적으로 대처 할 수 없다. 결과 DARPA를 비롯한 여러 기관에서 전송중인 액티브 패킷이 라우터에서 관리자의 정책을 담고 있는 코드를 실행할 수 있고 그 코드의 실행결과에 따라 라우터의 상태를 변경할 수 있는 액티브 네트워크 전반적인 구조를 제안하였다. 하지만 액티브 네트워크는 기존의 네트워크에서 부족했던 유연성 측면에서는 많은 이점을 갖지만 그 대신 이렇게 패킷에 담긴 액티브 코드가 실행되기 위해 액티브 노드의 자원에 접근해야하기에 정당하지 못한 패킷이 액티브 노드의 자원이나 시스템 자체에 악영향을 끼칠 수 있는 많은 보안상의 위협이 발생한다. 따라서 액티브 노드의 NodeOs단에 Crypto engine, Integrity Engine, Authentication Engine, Authorization Engine등을 비롯한 액티브 노드 인증 및 액티브 패킷/코드 인증 보안 모듈을 둠으로써 액티브 노드간 서로 안전한 협업적 관리를 할 수 있도록 한다. 즉 보안모듈에서는 액티브 노드가 받은 액티브 패킷이 올바른 액티브 노드로부터 왔는지 인증하고, 액티브 패킷과 코드가 오는 도중 변조 되지 않았는지 무결성을 제공하며, 액티브 패킷에 담고 있는 코드가 올바른지 확인하는 low-level 보안과 문제가 없을 경우 비로소 액티브 코드가 실행되는 high-level 보안을 제안하여 액티브 노드 간 안전한 통신을 가능케 하고자 한다.

즉 하나의 도메인 내의 액티브 노드들은 서로 협업적 보안관계를 통해 여러 가지 공격에 대응하여 액티브 노드를 보호하는 능력이 필요하다. 또한 라우터 등의 중간 노드의 시스템에 사용자의 실행 가능한 코드가 직접 접근해서 그 노드의 기능을 무력화시킬 수도 있다는 점에서 액티브 네트워크의 보안은 기존의 망보다 더욱 세심하게 고려되어야 한다.

2. 관련연구

2.1 FAIN[1]

FAIN은 2000년부터 UCL(University College London)이 주축이 되어 진행 중인 프로젝트로 액티브 노드를 기반으로 개방적이고 프로그램 가능하며 신뢰성 있는 액티브 네트워크 구조를 개발하는데 목적을 두고 있다. 이는 액티브 네트워크, 액티브 노드, 정책 기반의 네트워크 관리(policy-based network management)와 동적인 프로토콜 지원을 통해서 액티브 네트워크 서비스 제공을 목적으로 진행되고 있다.

FAIN에서는 보안을 라우터에 들어온 패킷은 기존의 패킷인 경우 다음 노드로 전송이 되고, 액티브 패킷인 경우 노드 운영 시스템을 거쳐 실행환경으로 가서 프로그램을 실행하게 되는 구조를 가지고 있다. 노드 운영시스템에는 보안 모듈과 자원 접근 모듈들이 포함되어 있어 들어온 패킷에 대해 보안 검사를 실시하고 통과된 패킷에 관해 할당된 자원을 분배하는 역할을 하고 있다.

3. 제안하는 방법

3.1 보안 요구사항

3.1.1 인증

액티브 네트워크에서 인증은 액티브 노드가 받은 액티브 패킷이 올바른 액티브 노드로부터 왔는지 확인하는 과정이다. 보통 인증은 요청을 원하는 개체와 요청을 받는 개체 간에 public/private key pair와 public key certificate를 가져야 하는 public key cryptography에 기반한다. 예를 들어 사용자는 사용자의 private key를 이용해 액티브 패킷에 디지털 서명을 하고 보낸다. 그러면 액티브 패킷을 받은 액티브 노드는 그것이 올바른 사용자로부터 왔는지 확인하기 위해, 즉 사용자의 public key를 입증하기 위해 public key certificate를 사용한다. 만약 유효하면, 액티브 노드는 패킷의 디지털 서명을 입증하기 위해 사용자의 public key를 사용한다. 따라서 인증을 위해 public key cryptography와 PKI infrastructure를 사용할 것이며 그것은 Crypto Engine과 Authentication Engine의 상호작용으로 이루어진다.

3.1.2 안전한 정책을 기반으로 하는 권한부여

권한 부여는 액티브 패킷에 대한 인증이 완료된 후 실제로 이 액티브 노드에서 액티브 패킷을 실행하게 할 건지 아닐 건지 실행에 대한 권한을 체크하는 것이다. 먼저 패킷이 액티브 노드에서의 실행을 위해 요구하는 자원에 대한 접근이 정당한 것인지 검증하고, 자원 사용에 대해 제한된 권한을 부여하게 된다. 각 액티브 노드는 정책 데이터베이스 (Policy DB)를 갖고 있으며, 정책 데이터베이스는 그 액티브 노드에서 실행 가능한 각 서비스 별 정책을 포함하고 있다. 해당 서비스의 정책에는 어떤 호스트와 사용자가 얼마만큼의 권한을 갖고 실행할 수 있는지 명시되어 있기 때문에 권한 부여 모듈은 패킷이 요청한 서비스에 해당하는 정책 데이터베이스를 참고하여 적절한 권한을 부여한다. 또한 액티브 노드는 동시에 많은 사용자의 코드가 실행 되야 하므로 우선권 부여를 고려한 정책적 관리가 필요하게 된다.

권한 부여에서는 패킷의 권한 부여 외에 노드내의 정책 데이터베이스를 관리할 수 있다. 즉 정책 데이터 베이스 내의 정책 정보들은 수정 혹은 삭제될 수 있으며, 새로운 서비스나 항목에 대한 정책이 정책 데이터베이스에 추가 될 수도 있다.

3.1.3 액티브 패킷 기밀성

기밀성은 전송되는 정보의 불법적인 노출을 방지하는 기술로 액티브 패킷을 암호화하여 전송함으로써 제공할 수 있다. 송신지가 자신의 비밀키로 암호화하여 전송하면 수신 측에서 공유하고 있는 비밀키로 복호화 하거나 송신지가 수신지의 공개키로 메시지를 암호화하여 전송하면 수신지가 자신의 비밀키로 메시지를 복호화한다.

3.1.4 액티브 패킷/코드 무결성

액티브 패킷과 코드는 네트워크 전송 중에 악의 있는 사용자에 의해 수정 변조되거나 재전송 될 수 있다. 따라서 액티브 패킷을 받은 액티브 노드는 액티브 패킷과 코드의 무결성을 검증해야 한다. 일반적으로 액티브 패킷은 전송 중 변하지 않는 정적인 부분과 전송 중 합법적으로 변할 수 있는 동적인 부분이 있기 때문에 액티브 노드들 간에는 end-to-end 아닌 hop-by-hop으로 무결성이 검증되야한다. Hop-by-hop으로 검증되기 위해서는 사용자는 a public key pair와 유효한 PK certificate를 가지고 있어야 하며, 각각의 액티브 노드는 그의 이웃과 공유하는 비밀 키를 가지고 있어야 한다. 각각의 패킷은 정적인 부분에 대해 전자 서명을 가지고 있어야 하며, 동적인 부분에 대해서는 MAC (Message Authentication Code)을 가지고 있어야 한다, 또한 anti-replay를 위해 timestamp라든지, 혹은 특정 값이 있어야 한다. 즉, 무결성은 패킷을 받은 액티브 노드가 MAC과 디지털 서명 및 anti-replay을 확인하는 것이다. 만약, 액티브 노드에서 패킷 무결성 확인 후 동적인 부분을 수정해야 할 일이 생긴다면 수정 후 결과 나온 새로운 MAC값을 다음 홉으로 보내면 된다. 이것은 integrity engine과 security facilities, authentication engine, crypto engine의 상호작용으로 이루어진다.

3.1.5 코드 입증

코드 입증이란 추가적인 보안 기능으로, 코드를 수행하기 전 신뢰성 없이 새롭게 도착한 액티브 코드의 수행을 막고자 하는 것이다. 만약 코드 입증이 실패면 그것은 코드는 그것에 대한 권한부여 및 우선권도 역시 신뢰할 수 없기에 실행될 필요가 없다는 것을 의미하므로 버려지고, 성공이면 권한 부여 엔진으로 가게 된다.

3.2 보안 구조 범위[2][3]

보안구조의 범위는 하나의 도메인 내에서의 액티브 노들들 간의 협업적 관리이며, [그림1]과 같이 2- level Security Framework을 가진다.

3.2.1 Low-Level Security Operation

액티브 코드가 실행되기 전 거치게 되는 보안 단계로 인증 및 액비트 패킷/코드의 무결성을 확인하는 작업이다.

3.2.2 High-Level Security Operation

3.2.1에서 기술된 Low-Level Security Operation에서 패킷이 무결성이 보장된 이후 실제 액티브 코드가 수행되는 보안 단계이다.

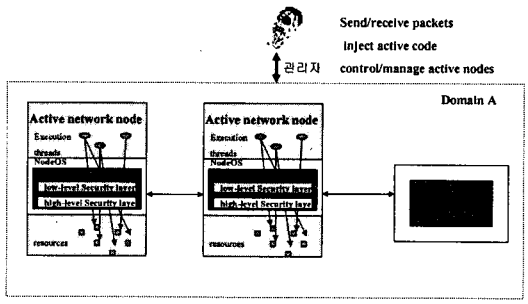


그림 1 협업적 관리를 위한 보안구조

3.3 액티브 노드 보안 모듈

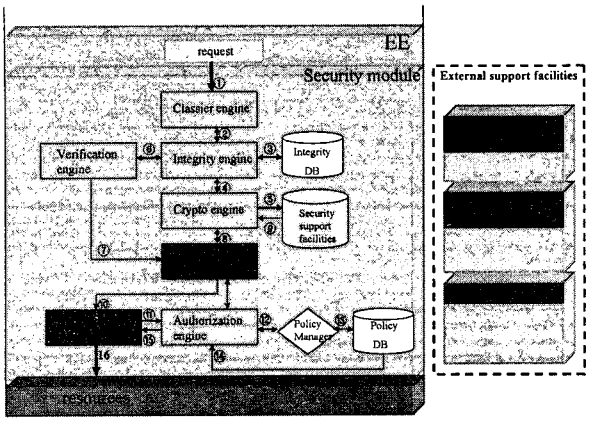


그림 2 액티브 노드 보안 모듈

3.3.1 Crypto Engine

symmetric encryption/decryption이나 asymmetric encryption/decryption, hashing과 같은 다양한 암호화 알고리즘을 수행하는 엔진으로 기타 다른 보안 엔진에서 사용자를 인증하거나 액티브 패킷/코드의 무결성 검증시 사용된다.

3.3.2 Integrity engine

Crypto engine상의 암호화 알고리즘을 이용하여 액티브 패킷과 액티브 코드의 무결성을 확인하는 엔진이다.

3.3.3 Verification engine

Crypto engine상의 암호화 알고리즘을 이용하여 코드를 수행하기 전 신뢰성 없이 새롭게 도착한 액티브 코드의 수행을 막는 엔진이다.

3.3.4 Authentication engine

Crypto engine상의 암호화 알고리즘을 이용하여 액티브 노드가 받은 액티브 패킷이 올바른 액티브 노드로부터 왔는지 확인하는 엔진이다.

3.3.5 Authorization engine

액티브 패킷에 대한 인증이 완료된 후 액티브 패킷에 대한 수행 여부의 권한을 체크하고 부여하는 엔진이다.

3.3.6 Policy DB

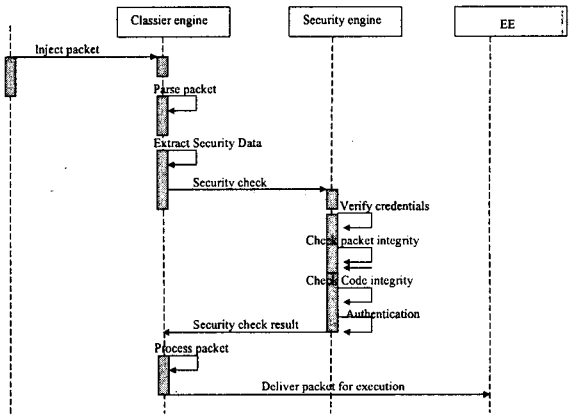
액티브 노드 안에서 누가 무엇을 할 수 있는지에 관한 보안 정책을 담고 있는 데이터베이스이다.

3.3.7 Policy Manager

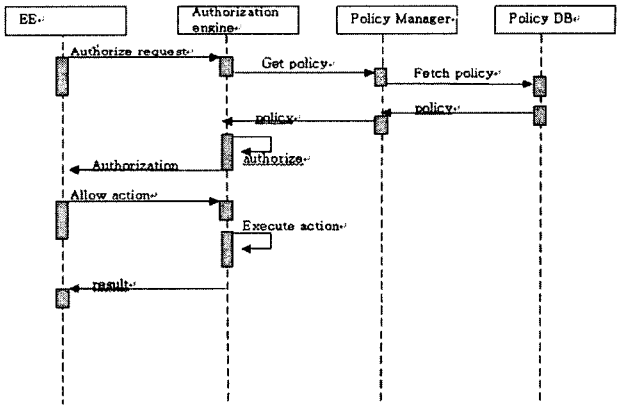
Authorization engine에 의해 요청되면 요청된 서비스와 관련된 보안 정책들을 정책데이터베이스에서 찾고 그 결과를 돌려주는 매니저다. 또한 노드내의 정책 데이터베이스에 새로운 서비스나 항목에 대한 정책을 추가, 수정 삭제 할 수 있도록 해준다.

3.4 보안 구조내의 작동

3.4.1 Low-Level security operation



3.4.2 High-Level security operation



4. 결론

하나의 도메인 내의 액티브 노드들의 협업적 보안관계를 통해 여러 가지 공격에 대응하여 액티브 노드를 보호하였고 또한 라우터 등의 중간 노드의 시스템에 사용자의 실행 가능한 코드가 직접 접근해서 그 노드의 기능을 무력화시킬 수도 있다는 점에서 액티브 네트워크의 보안을 기존의 망보다 더욱 강화하였다.

5. 참고문헌

[1] Initial Active Network and Node Architecture-FAIN Deliverable D2.
[2] 3GPP TS 32.102 v6.3.0 (2004-06) TM Architecture.
[3] IEEE 1520 Standards, Initiative for Programmable Network Interfaces.