객체지향 미들웨어 환경하에서 응용화된 침입감내시스템에 관한 연구

김영수*°,최흥식**

*국민대학교 정보관리학과, **국민대학교 정보관리학과

Study on the Intrusion Tolerance System Applied in the Object Oriented Middleware Environment

Kim, Young Soo, Choi, Heung-Sik

Kookmin University, Kookmin University

E-mail : experkim@dreamwiz.com, hschoi@kookmin.ac.kr

요 약

오늘날 대부분의 컴퓨터 시스템에 대한 사이버 공격은 특정 어플리케이션이 제기능을 발휘하지 못하도록 하기 위한 의도로 행해진다. 이러한 사이버 공격을 방어하기 위한 현행 정보보호시스템은 어플리케이션으로의 접근을 보호하는 기능을 갖는 형태로 하드웨어, 네트워크, 운영체제등의 인프라를 보호하기 위한 시스템이 대부분이다. 본 논문은 사이버 공격에도 불구하고 어플리케이션이 서비스를 지속할 수 있는 침입감내시스템을 제안하고 설계 구현하였다. 제안 시스템은 어플리케이션의 기능과 침입감내 기능을 분리하고 다양한 보안 메커니즘을 통합하고 있는 미들웨어 기반의 시스템이다. 일상 생활에 많은 영향을 미치는 중요한 서비스인 경우에는 시스템의 기능 중 서비스의 지속성이 가장 중요하다. 제안한 침입감내 시스템은 향후 침입감내기능이 필요한 국방, 의료, 금융 시스템과 같은 사회기반시스템의 구축에 많은 부분이 활용될 것으로 생각한다.

1. 서론

대부분의 분산 정보 시스템은 네트웍 폭주로 인한 서비스의 장애와 가용성를 제한 받는 취약한 형태를 보이고 있다[1]. 우리의 생활에 많은 영향을 미치는 의료시스템, 금융시스템과 국가의 안위와 직결되는 국방 시스템등이 점차 네트워크를 기반으로 한 분산시스템에 의존해가면서 이러한 취약성으로 인해 발생되는 서비스의 중단은 사회의 혼란을 야기하므로 시급히 해결해야 할 과제가 되고 있다[2].

서비스의 중지를 가져오는 취약성의 문제를 해결하고자 제안된 시스템이 침입 감내 시스템이다. 결함 허용 시스템이 시스템과 소프트웨어의 우발적 결함에 대하여 관심을 가지는 반면 침입감내 시스템은 우발적 범주의 결함 이외에도 악의적인 행위에 의하여 발생하는 결함 즉 침입에 관심을 가진다는 것이다[3]. 침입 감내 시스템은 우발적 사고 또는 악의적 공격에 대한 해결책을 찾을 때까지 적절한 대응을 통해 서비스의 품질 저하를 방지하면서 일정시간 동안 중요한 서비스를 지속적으로 제공하는 것을 목표로 하는 시스템이다.

이러한 침입감내는 보안 메커니즘과 적응 메커니즘을 통해 제공될 수 있다[4][5]. 보안 메카니즘을 사용하는 어플리케이션도 사이버 공격에는 취약점을 갖을수 있는 반면 적응 메커니즘을 사용하는 어플리케이션은 사이버공격에도 불구하고 일정시간동안 생존할 수 있다. 따라서 침입감내 기능을 갖는 어플리케이션은 보안 메커니즘과 적응 메커니즘을 사용하여 구현할 수 있다.

억세스 제어나 암호화와 같은 보안 메커니즘은

우발적 또는 악의적인 공격으로부터 어플리케이션을 보호함으로써 침입감내 기능을 강화할 수 있고 대역폭 관리나 중복관리를 사용하는 적응 메커니즘은 시스템의 상태 변화에 적응함으로써 침입감내 기능을 제공할 수 있다[6].

본 논문은 다양한 보안 메커니즘과 적응 메커니즘의 서비스를 미들웨어 상에서 구현하여 어플리케이션의 기능과 분리된 형태로 침입감내를 구현하였다. 본 논문은 다음과 같이 구성된다. 2장에서는 코바의 분산객체모델과 보안모델을 분석하여 침입감내 보안 모델을 제안하였다. 3장에서는 침입감내 메커니즘의 원리와 구현전략에 대해 설명하고 침입감내기술과 구현 방안을 도출하였다. 4장에서는 침입감내 보안모델과 요소기술을 반영하여 시스템을 분석 설계하였고 5장에서는 개발한 침입감내 모듈을 미들웨어 형태로 사용하여 침입 감내 어플리케이션을 구축할 수 있음을 확인하였다.

2. 보안 모델

1) 분산객체 모델

오늘날 네트워크 기반의 응용 시스템은 분산객체 기술을 사용하여 개발되고 있다. 분산정보시스템을 구성하는 소프트웨어는 네트워크상에 분산되어 있는 객체들로 구성되어 있고 클라이언트 객체는 어플리케이션의 기능을 달성하기 위하여 원격에서 실행되는 종복 객체의 오퍼레이션을 호출한다[7].

코바 분산객체 모델은 아래 그림 1과 같이 클라이언트 객체는 마치 로컬 객체인 것처럼 서버상의 원격객체의 메서드를 호출한다.

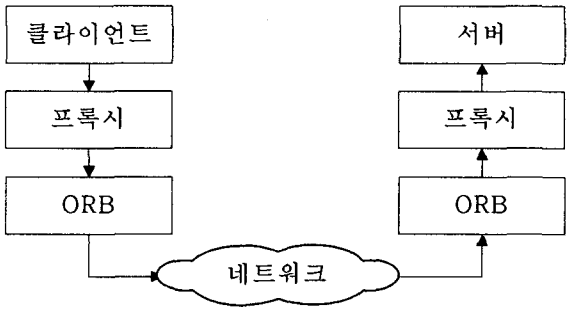

그림 1. 분산객체모델

스터브 또는 프록시 객체는 원격 메서드 호출를 마샬링 한 후에 로컬 ORB에게 전달한다. ORB는 IIOP 메시지를 사용하여 메서드 호출을 종복 객체가 있는 원격 ORB에게 전송한다. 원격 ORB는 언마샬링을 수행하는 스켈리톤에게 메서드 호출을 전송하고 스켈리톤은 메서드 호출을 구현객체에게 전송한다. 메서드의 수행 결과값은 그 역으로 전송된다. 분산객체 모델은 다양성, 이종성, 상호호환성 같은 복잡성을 숨기고 컴포넌트의 기능적 인터페이스만을 보여준다.

2) 분산객체 보안 모델

코바 분산 객체 시스템은 미들웨어상에서 정책기반의 보안 메커니즘을 사용함으로서 객체의 기밀성, 무결성, 가용성을 보증한다[8]. 보안정책은 그룹화 메커니즘인 도메인, 특권, 사용권한을 사용하여 일괄적으로 설정한다. 즉 보안관리자는 객체를 도메인에 포함시키고 객체들의 그룹에 대하여 일률적으로 보안정책을 설정하고 사용자들을 동일 특권속성으로 그룹화하여 사용자 그룹에 대하여 보안정책을 실시한다. 그리고 객체의 메서드에 대해서는 동일한 사용권한을 부여함으로써 메서드를 그룹화하고 그룹에 대하여 보안정책을 적용한다. 그림 2는 분산객체 보안 모델을 묘사하고 있다.

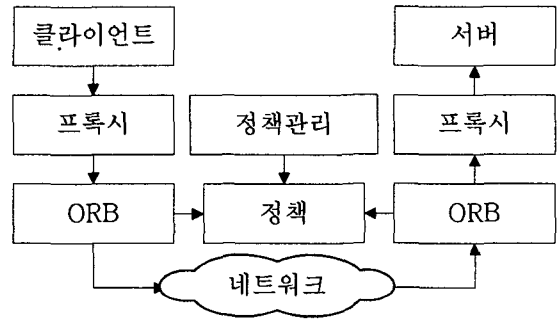

그림 2. 분산객체 보안모델

3)침입감내 보안모델

국방, 의료, 금융 시스템과 같은 어플리케이션은 보안, 신뢰성 그리고 실시간 성능과 같은 요구사항을 필요로 한다. 분산객체 미들웨어는 서비스 품질을 명기하고 측정하고 통제하는데 필요한 세부내용을 숨기고 서비스 품질의 변화에 적응할 수 있는 시스템 개발 기능을 제공하지 않기 때문에 이러한 요구사항들을 제공하지 못한다. 따라서 중요한 서비스를 제공하는 어플리케이션 개발자는 미

들웨어를 기반으로 침입감내기능을 갖는 어플리케이션을 개발해야 한다.

본 논문에서는 침임감내 보안 모델로 그림 3과 같은 분산객체 모델을 확장한 형태를 제안한다.

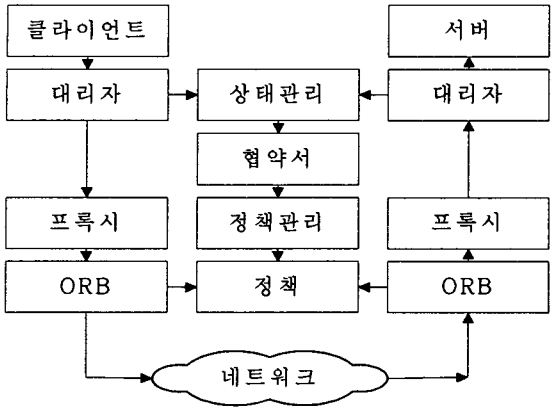

그림 3. 침입감내 보안 모델

분산객체 미들웨어에 의하여 숨겨져 있으나 서비스 품질을 명기하고 측정하고 통제하는데 필수적인 세부사항과 어플리케이션과의 브리지 역할을 하는 기능을 추가하였다. 서비스 요구사항과 서비스 범주는 협약서(Contract)라고 명명한 데이터 구조체속에 유지한다. 대리자(Delegate)는 시스템의 상태를 파악하고 협약서내의 서비스 수준과의 비교를 통해 적절한 적응 객체의 메서드를 수행시킨다. 상태 관리 객체는 보안 메커니즘의 상태를 측정하는 기능을 수행한다.

제안 시스템은 서비스 품질을 관리하기 위한 프레임웍을 제공하는데 목적을 두고 있다. 따라서 신뢰성과 대역폭 그리고 보안등의 서비스 품질을 관리하는 중복관리 시스템. 대역폭 관리 시스템, 침입탐지시스템과 같은 메커니즘은 제안 시스템의 통합된 부분이 아니다. 따라서 서비스 품질 요구사항에 따라 필요한 메커니즘을 제안된 시스템내에 통합하여 사용할 필요가 있다.

3. 침입감내 메커니즘

1) 침입감내 응용 원리

침입감내 메커니즘은 적응성, 중복성, 감시성, 다양성의 원리를 기반으로 시스템에 대한 불법적인 침입이나 공격에 대한 대책마련을 위하여 시도되는 새로운 정보보호기술이다. 침입감내기술은 침입방지시스템과 침입탐지시스템을 뚫고 침입하는 고도의 기술을 가진 침입자들에 대한 대책을 마련하기 위하여 다양한 메커니즘을 사용한다.

-적응성 원리

침입감내 기능을 제공하는 어플리케이션은 침입이나 바이러스 공격과 같은 악의적 행위로 발생되는 시스템의 상태 변화에 적응 할 수 있어야 한다.

- 중복성 원리

침입 감내 어플리케이션은 서버나 사이트와 같은 자원을 중복시키고 서버의 다운이나 네트워크의 혼잡과 같은 경우에 동일한 서비스를 제공하는 대체 서버나 사이트로 전환할 수 있어야 한다.

- 감시성 원리

침입 감내 어플리케이션은 대역폭 관리시스템, 중복관리 시스템 ,침입차단 시스템, 침입방지 시스템으로부터 얻은 시스템의 상태 데이터를 적응에 사용할수 있어야 한다.

- 다양성 원리

동일한 컴퓨팅 자원은 동일한 시간에 동일한 문제점을 보유함으로 이를 방지하기 위하여 동일한 서비스를 제공하는 상이한 운영체제나 상이한 알고리즘을 사용하는 객체구현과 같은 다양한 컴퓨팅 자원을 사용하여야 한다.

이러한 메커니즘은 독립적으로 사용 될 수 있으나 상호 보완적으로 사용함으로써 침입감내능을 향상시킬 수 있다.

2) 게이트웨이 프로토콜

게이트웨이 컴포넌트는 클라이언트 ORB와 서버의 ORB간에 송수신하는 IIOP 메시지를 인터셉트하여 인터넷 시스템과 중복관리 시스템 그리고 대역폭 관리 시스템의 전송 프로토콜로 변환하는 기능을 수행하거나 인증 시스템이나 억세스 제어시스템을 이용하여 적절한 적응과 통제를 하는 기능을 수행하도록 설계하였다.

게이트웨이 프로토콜 구성도는 그림 4와 같다.

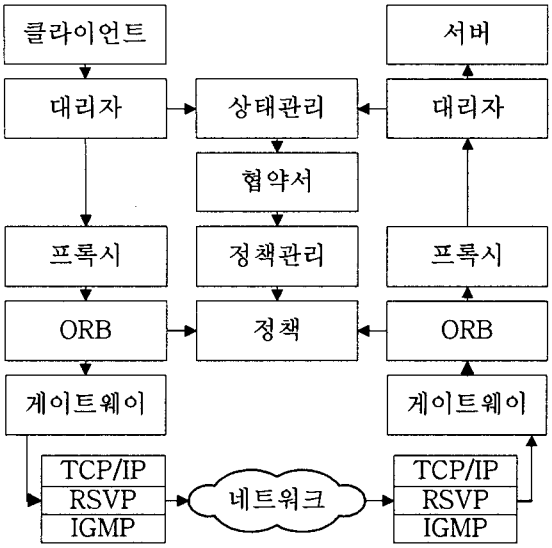

그림 4. 침입감내 게이트웨이

클라이언트가 서버 구현객체의 메서드를 호출할 때 게이트웨이 컴포넌트는 IIOP 메시지를 인터셉트하여 인증서버로 전송하고 리턴되는 인증 데이터를 기반으로 적절한 인증을 실현한다.

3) 보호에 의한 침입감내

네트워크를 기반으로 하는 분산정보 시스템의 보호를 위한 최초의 방어선은 억세스 제어이다. 코바에서 정의하고 있는 정책 지향적인 억세스 제어를 상용 미들웨어는 구현하고 있다.

그림 5와 같이 ID와 특권 내용을 담고 있는 신임장 객체와 정책 객체 그리고 객체 참조에 의해서 서비스 품질에 대한 보안을 실현한다.

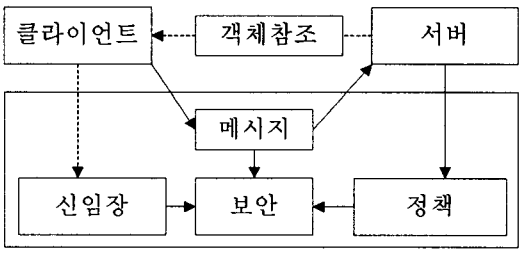

그림 5. 침입감내 보안메커니즘

4) 적응에 의한 침입감내

침입감내 시스템은 어플리케이션에서 이용가능한 서비스 품질을 감시하고 이를 변화시킬 수 있어야 한다. 어플리케이션은 시스템의 서비스 품질이 저하되는 경우에 시스템의 고장을 유발하는 대신 적응할 수 있어야 한다. 어플리케이션의 서비스 품질을 명기하고 감시하고 통제하고 적응시킬수 있는 침입 감내 기능을 미들웨어상에 구현함으로써 어플리케이션의 기능과 분리되도록 해야 한다.

이는 분산정보시스템를 개발하기 위해서는 클라이언트와 구현객체를 개발하는 어플리케이션 개발자와 ORB와 같은 미들웨어 개발자 그리고 서비스 품질 개발자로 역할을 분담하여 수행할 수 있다. 사이버 공격에도 불구하고 어플리케이션이 서비스를 지속할 수 있도록 몇가지 적응전략을 생각해 볼 수 있다. 첫째 아무것도 하지 않는다. 둘째 사이버 공격이 불가능한 시스템을 구축한다. 셋째 사이버공격을 예측하고 대처할 수 있는 어플리케이션을 개발한다. 넷째 사이버공격에 적응할 수 있는 메커니즘을 사용하여 어플리케이션을 개발한다.

첫 번째 전략은 사이버 공격이 발생하는 경우에 시스템의 고장을 가져오게 된다. 둘째와 셋째는 사이버 공격의 가능성을 예측할수 없기 때문에 타당하지 않다. 네 번째 전략이 가장 타당하다 그림 6은 적응 메커니즘을 사용하여 구현한 침입감내시스템을 보여주고 있다.

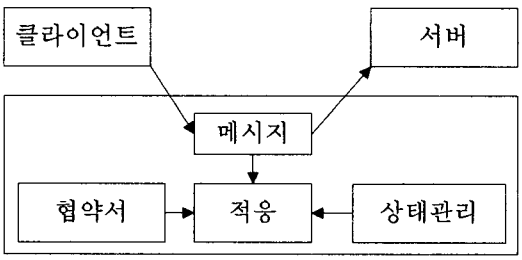

그림 6. 침입감내 적응메커니즘

4. 침입감내 시스템의 모델링

침입감내 기능은 시스템과 외부환경 측면에서 묘사되어질수 있다[9]. 지속적으로 제공되어야 할 중요한 시스템의 기능을 서비스 워크플로우 기반으로 표현하고 외부환경에 적응할수 있는 기능을 침입 워크플로우 기반으로 기술함으로써 침입감내 기능을 도출한다[10].

즉 외부의 사이버 공격에도 불구하고 지속적으로 제공되어야 할 시스템의 필수 서비스의 정의와

설계 그리고 이러한 필수 서비스에 대한 외부 침입 명세서와 분석을 바탕으로 침입감내시스템을 구현해야 한다.

침입감내 어플리케이션 개발 과정은 그림 7과 같이 필수서비스와 이를 공격하는 침입의 분석과 설계를 바탕으로 구현 한다.

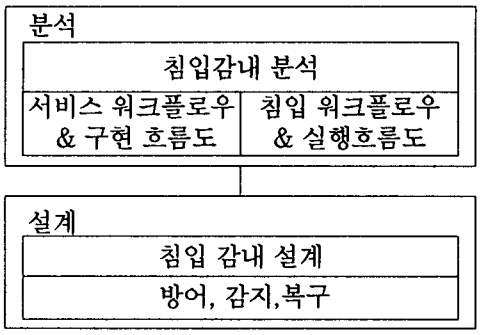

그림 7. 침입감내 시스템의 모델링

필수 서비스는 워크플로우의 형태로 정의되어질 수 있고 워크플로우는 일련의 시스템 아키텍처 컴포넌트를 구성요소로 갖는 필수 서비스 구현 흐름도로 변환되어질 수 있다.

이러한 구현흐름도의 각 컴포넌트에 대해서 침입 패턴을 정의한다. 침입패턴으로부터 침입 워크플로우를 작성하고 침입 워크플로우로부터 침입 실행 흐름도를 작성한다[11].

필수서비스를 표현하고 있는 워크플로우는 작업, 정보. 의사결정, 흐름으로 구성된다. 실행흐름도는 워크플로우상의 작업을 수작업 또는 자동화를 위한 아키텍처 컴포넌트로 대체하고 정보는 데이터로서 표현된다. 의사결정은 데이터의 평가에 바탕을 두도록 변환된다. 실행 흐름도의 구성요소의 취약성을 공격하는 침입 분석과 설계를 수행한다.

그림 8과 같이 각각의 침입시나리오에 대하여 방어, 감지, 복구메커니즘을 기술한다.

구분	전략	메커니즘
침입시나리오	방어	인 증
	감지	침입탐지
	복구	중복

그림 8. 침입감내 전략 모델링

5. 침입감내시스템의 구현

1) 응용시스템

침입감내시스템의 구현 목적은 미들웨어 계층에서 다양한 보안 메커니즘의 통합방법과 적응 메커니즘의 활용 방법을 제시하는데 있다. 이를 위해 침입감내기능을 갖는 재고관리시스템을 단순하게 구현하였다.

클라이언트는 로그인 기능, 재고상품의 요청처리 기능 그리고 로그아웃기능을 갖고 서버는 재고상품DB와 재고관리서비스를 제공하도록 구현하였다. 그림 9는 재고관리 시스템의 구성도를 보인다.

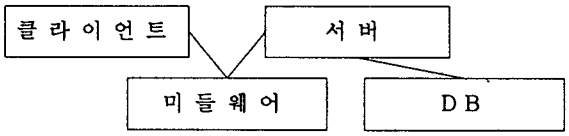

그림 9. 응용시스템

2) 응용화된 침입감내시스템

서비스의 지속성을 중요시하는 분산 시스템의 구축시 상용 어플리케이션과 운영체제를 일반적으로 사용한다,. 이는 어플리케이션 서버의 개발보다는 상용제품을 사용하는 것이 노력과 비용이 덜 소요되기 때문이다.

그러나 상용 어플리케이션 서버는 많은 보안상의 취약성을 가지고 있다. 이러한 취약점을 해결하기 위하여 어플리케이션 서버와 클라이언트 사이에 보안기능을 갖는 소프트웨어 계층을 인터페이스로 사용한다. 즉 인터페이스 계층에서 클라이언트와 서버간의 메시지를 인터셉터하여 보안을 실현한다[12].

그림 10은 어플리케이션 서버의 API를 캡슐화한 래퍼를 사용하여 보안을 실현하고 있다. 따라서 클라이언트는 래퍼를 이용하여 서버의 서비스를 제공 받으므로 송수신되는 모든 메시지를 래퍼는 감시하고 변경할 수 있다. 보안기능을 갖도록 래퍼를 구성함으로써 상용제품이 보안성을 갖도록 할 수 있다.

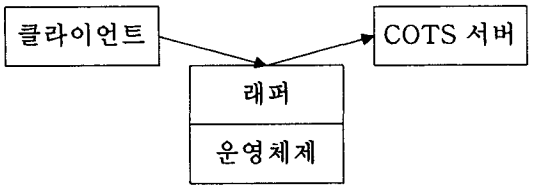

그림 10. 소프트웨어 래퍼

침입감내 기능을 갖도록 변경한 응용시스템의 구성도는 그림 11과 같다.

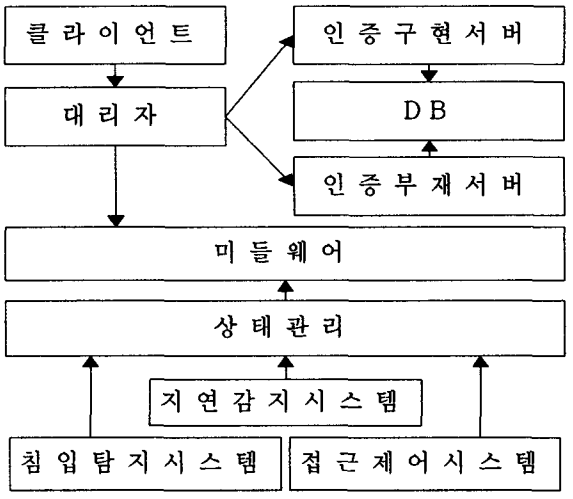

그림 11. 침입감내시스템시스템

보안 메커니즘으로는 침입탐지시스템과 접근제어을 사용하였다. 침입탐지시스템과 접근제어시스템의 인터페이스 역할을 하는 상태관리 객체는 IDL과 래퍼객체를 사용하여 구현하였고 서비스 품질수준을 시간지연, 접근위반, 침입탐지 모드의 형태로 협약서 객체에 정의하였다.

적응 메커니즘으로는 인증기능을 갖는 서버와 인증기능이 없는 서버를 중복하여 사용하였다. 대리자 객체의 기능은 그림 12와 같다.

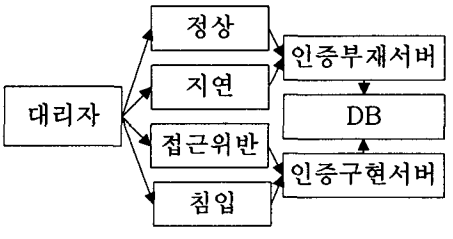

그림 12. 대리자 객체의 기능

대리자객체는 서비스품질수준이 정상 모드에 해당되는 경우와 시간지연 모드인 경우에는 인증기능이 없는 서버에게로 요청 메시지를 전달하여 처리하도록 하였다. 그러나 접근위반과 침입 모드인 경우에는 인증기능을 수행하는 서버에게로 요청메시지를 전송하여 처리하도록 구현하였다.

3) 응용계층과 미들웨어계층상의 구현비교

침입감내 기능을 응용계층상에서 구현할 때의 문제점은 비즈니스 로직과 침입감내기능이 분리되지 않음으로써 프로그램의 변경과 확장이 쉽지 않고 유지보수가 어렵다는 단점이 있다.

그림 13은 응응계층에서의 침입감내 구현에 대한 구성도이다.

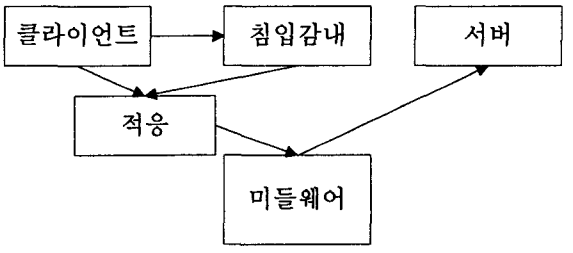

그림 13. 응용계층의 침입감내 구현

반면 그림 14는 미들웨어상에서 침입감내기능을 구현하는 구성도를 보여주고 있다.

장점은 침입감내기능을 비즈니스 로직과 분리 구현할 수 있어 어플리케이션 개발자와 침입감내 구현 개발자로 역할을 분담하여 개발 작업을 할 수 있음으로 우수한 보안성을 갖는 어플리케이션을 구축할 수 있다.

또한 침입감내 구현 개발자를 위한 개발툴을 제작하여 제공함으로써 소프트웨어의 생산성을 증가시킬 수 있다. 그리고 구현한 침입감내 기능을 미들웨어의 형태로 사용할 수 있어서 어플리케이션을 침입감내 미들웨어에 플러그인함으로써 침입감내 기능을 쉽게 구현할 수 있고 침입 감내기능을 표준화함으로써 많은 상이한 침입감내시스템을 통합할 수 있는 잇점을 제공한다.

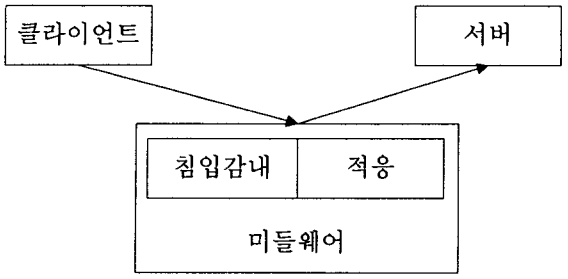

그림 14. 미들웨어계층의 침입감내 구현

6. 결론

최근에 보고되는 침해 사고들은 정상적인 서비스를 제공할 수 없도록 시스템의 애플리케이션을 공격하는 형태를 취하고 있다. 기존 정보보호 접근법은 애플리케이션의 공격을 방지하기 위하여 하드웨어, 운영체제, 네트워크와 같은 인프라를 안전하게 보호하는 것이다.

이러한 보안 메커니즘은 시스템의 공격으로부터 애플리케이션을 보호함으로서 서비스를 지속적으로 제공할 수 있는 반면 침입감내 메커니즘은 시스템에 대한 악의적 공격이 발생하여도 공격으로 인한 변화에 애플리케이션을 적응시킴으로써 중요한 서비스를 지속적으로 제공하는 것이다. 보안 메커니즘을 애플리케이션에 적용함으로써 침입감내를 제공할 수 있으므로 보안과 침입감내는 상호 배타적이 아니라 보완관계를 가지고 있다.

본 논문은 시스템에 대한 공격이 발생하더라도 정상적인 서비스를 제공할 수 있도록 미들웨어와 연계되는 침입감내 애플리케이션의 구축 방안을 연구하였다. 침입감내시스템의 구축에 액세스 제어, 패킷 필터링, 그리고 침입탐지 메커니즘등의 보안 메커니즘을 적용하고 중복관리시스템과 대역폭관리시스템과 같은 침입감내 메커니즘의 응용을 통해 침입 감내를 제공하도록 하였다. 이러한 다양한 메커니즘의 통합을 위하여 어댑터 인터페이스의 정의와 래퍼 클래스의 구현 방안을 제시하였다.

애플리케이션 코드를 사용하여 침입감내 메커니즘을 구현하는 경우에는 애플리케이션의 기능과 침입감내 코드가 분리되지 않음으로 인해 재사용의 제한을 받게 된다. 따라서 미들웨어 계층에서 침입감내 메커니즘을 구현해야 한다. 이는 침입감내 애플리케이션 개발자가 서비스 수준과 서비스 요구사항을 명세화하고 침입감내 메커니즘으로부터 얻은 정보를 분석하고 평가한 후 변화에 적응시키려는 메서드를 기술하면 코드 생성기에 의해 침입감내 코드를 자동생성할 수 있음으로 소프트웨어 생산성을 향상시킬 수 있다. 감내하고자 하는 대상으로는 분산 애플리케이션 객체의 침해에 관한 것을 다루고 있다.

본 논문에서 제안하고 있는 침입감내시스템은 급속하게 늘어가고 있는 침해사고에서 애플리케이션이 생존할 수 있도록 함으로써 시스템의 신뢰성과 안정성을 높여 줄 수 있고 지나친 보안시스템의 구축에 따른 사용자의 불편을 감소시킬 수 있을 것으로 기대된다.

[참고문헌]

[1]Scambray, J., S. McClure, G. Kurtz, Hacking Exposed: Network Security Secrets & Solutions (Second Edition), McGraw-Hill, 2001

[2]Schneider, B. Secrets and Lies: Di-gital Security in a Networked World. New York NY: John Wiley & Sons, August 2000.

[3] Hartman, J. and Evans, W., Fault tolerant application enhanced network-Project A project under the DARPA Fault Tolerant Networks Program. Homepage. Internet URL http://www.cs.arizona.edu/ftn, 2000.

[4] Loyall, J. P., Pal, P. P., Schantz, R. E., and Webber. F., Building adaptive and agile applications using intrusion detection and response. In Proceedings of the ISOC Network and Distributed Systems Security Conference, Februrary 2000.

[5] Cardei, M., Cardei, I., Jha, R., and Pavan. A., Hierarchical feedback adaptation for real time sensor-based distributed applications. In Proceedings of Middleware 2000 (LNCS 1795), pages 415-435. Springer-Verlag, 2000.

[6] Loyall, J. P., Pal, P. P., Schantz, R. E. and Webber. F., Building adaptive and agile applications using intrusion detection and response. Im Proceedings of the ISOC Network and distributed Systems Security Conference, Februrary 2000.

[7] Li, B. and Lahrstedt. K., Qualprobes:middleware qos profiling services for configuring adaptive applications. In Proceedings of Middleware 2000 (LNCS 1795), Pages 256-272. Springer-Verlag, 2000.

[8] Pal, P., Loyall, J., Zinky, J., Shapiro, R. and Megquier, J., Using qdl to specify qos aware distributed(quo) application configuration. In Proceedings of The 3rd IEEE International Symposium on Object-oriented Real-time distributed Computing (ISORC 00), March 2000.

[9] Hevner, A. Linger, R., Sobel, A., and Walton, G. Specifying Large-Scale, Adaptive Systems with Flow-Service-Quality(FSQ) Objects, Proceedings of 10th OOPSLA Workshop on Behavioral Semantics, Tampa, October 2001.

[10]Linger, R., Pleszkoch, M., Walton, G., and Hevner, A Flow-Service-Quality Engineering: Foundations for Network System Analysis and Developement, Software Engineering Institute, Carnegie Mellon University, CMU/SEI-2002-TN-019, July, 2002.

[11] Hayes, J., et al. "Workflow Interoperability Standards for the Internet." IEEE Internet Computing 4, 37-45. 3(May/June 2000).

[12] Badger, L., Generic software wrappers. Internet URL http://www.pgp.com/research/nailabs/secure-execution/wrappers-overview.asp. 2000